

***Títol:*** Disseny i implementació d'un sistema bluetooth per al tractament de l'enuresi.

***Volum:*** 1 de 1

***Alumne:*** Oriol Ciurana Adell

***Director/Ponent:*** Enric Xavier Martin Rull

***Departament:*** ESAII

***Data:*** Juny de 2010



---

## **DADES DEL PROJECTE**

*Títol del Projecte: Disseny i implementació d'un sistema bluetooth per al tractament de l'enuresi.*

*Nom de l'estudiant: Oriol Ciurana Adell*

*Titulació: Enginyeria Tècnica Informàtica de Sistemes*

*Crèdits: 22.5*

*Director/Ponent: Enric Xavier Martin Rull*

*Departament: Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII)*

---

## **MEMBRES DEL TRIBUNAL (nom i signatura)**

*President: Pere Marés Martí*

*Vocal: Juan Carlos Cruellas Ibarz*

*Secretari: Enric Xavier Martin Rull*

---

## **QUALIFICACIÓ**

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:*

---





# **Disseny i implementació d'un sistema bluetooth per al tractament de l'enuresi**

Projecte PPStop

Autor: Oriol Ciurana Adell  
Director: Enric Xavier Martin Rull  
Codirector: Josep Peguerols Valles

Departament d'Enginyeria de Sistemes, Automàtica  
i Informàtica Industrial (ESAI)

Barcelona, juny de 2010



# Agraïments

M'agradaria agrair a totes aquelles persones que d'una manera o altra m'han ajudat durant aquests mesos en la realització del projecte.

Agrair a l'Enric Xavier Martin pel suport rebut i a en Josep Peguerols la seva paciència, ajuda i confiança que m'ha mostrat des del primer dia. A tots els companys de *SerTel* per estar sempre disposats a donar-me un cop de mà, i més especialment a Mario Viktorov per la seva col·laboració en el projecte i per fer més disteses les estones de despatx. A Òscar Bergua de *IDOM* i a Santi Navarro, Dr. Jaume Payerols i Dr. Santiago Garcia-Tornel de l'*Hospital Sant Joan de Déu* de Barcelona per creure en aquest projecte i donar-me l'oportunitat de participar-hi. També a en Bernat Feixas d'*Electrònica Feixas*, perquè malgrat van sorgir alguns problemes sempre ha estat allí.

No puc deixar d'agrair també el suport dels meus amics, que m'han aguantat i per alguna estranya raó segueixen fent-ho tots els dies. Gràcies a tots els amics de Reus per no donar-me definitivament per desaparegut. Gràcies als amics *penyates*, als *Mukangendians*, als compis de la *FIB* i als de l'*ETSETB*. Gràcies als companys de *DAT*, als pacos de *Telecogresca* i en general a tots els de l'*Omega* pels grans moments que hem passat. Gràcies a l'Eric Marck per entendre que aquestes darreres setmanes la guitarra no era una prioritat. I gràcies a tots aquells que em deixo, que segur que són molts!

Ja per acabar, agrair el suport i paciència de tota la meva família. Gràcies pare per estar tant orgullós de mi. M'agradaria finalment deixar unes línies per a la meva mare i al meu germà, mai tindrè paraules suficients per agrair-vos tot el que heu fet per mi: gràcies per aguantar el meu estrès continu, el meu mal humor, el meu absentisme, la meva estranya manera de fer les coses, gràcies per haver-me donat l'oportunitat de cursar uns estudis universitaris i haver-me guiat pel bon camí, i sobretot gràcies per les incomptables empentes morals que m'heu donat. Segurament us alegrareu tant o més que jo de que per fi hagi acabat la carrera. Sense vosaltres, no ho hauria aconseguit.



# Resum

En aquesta memòria de Projecte Final de Carrera trobareu la documentació referent a tot el procés de disseny i d'implementació d'un sistema inalàmblic Bluetooth per al tractament del trastorn de enuresi en la seva forma passiva. L'objectiu principal del projecte és l'obtenció d'un prototip funcional de dispositiu electrònic que a partir de la captació d'un event extern aconsegueixi —mitjançant Bluetooth— activar una alarma sonora en un terminal mòbil dotat d'una aplicació especialment programada per a aquest efecte. A més a més, aquesta aplicació ha de ser capaç de transmetre —també mitjançant Bluetooth— el registre d'alarmes a una aplicació de seguiment per al seu posterior anàlisi i seguiment del tractament.

El procés de disseny que s'ha documentat engloba la presa de requisits del sistema a desenvolupar —tant funcionals com tècnics— i la presa de decisions sobre com han de ser els diferents elements que intervenen en el sistema —dispositiu electrònic, aplicació del terminal mòbil, protocols de comunicació—. El procés d'implementació que s'ha documentat engloba la programació de l'aplicació J2ME del terminal mòbil, la programació de l'aplicació J2SE per al seguiment del tractament així com la programació d'un simulador i posterior programació del microcontrolador —unitat de control— del dispositiu electrònic.

Durant el transcurs de la memòria es descriuen totes aquelles tecnologies —J2ME, Bluetooth, etc— que s'han utilitzat, i es fa un anàlisi de les línies futures per a la preparació comercial del sistema. Els resultats obtinguts en aquest projecte estan sent utilitzats per a l'elaboració d'un producte comercial —anomenat PPStop, llegeixi's Pi-Pi-Stop— per al tractament de l'enuresi per part del departament I+D de l'Hospital Universitari Sant Joan de Déu de Barcelona, centre especialitzat en aquest tipus de trastorn.



# Continguts

<b>Llistats</b>	<b>x</b>
Llista de figures . . . . .	xii
Llista de taules . . . . .	xiii
Llista de codis . . . . .	xvi
<b>1 Introducció, motivació i objectius</b>	<b>1</b>
1.1 L'enuresi i el seu tractament . . . . .	1
1.2 Motivació . . . . .	2
1.3 Objectius a assolir . . . . .	4
1.4 Estructura de la memòria . . . . .	5
<b>2 Disseny del sistema</b>	<b>9</b>
2.1 Requisits del sistema . . . . .	9
2.1.1 Requisits funcionals . . . . .	9
2.1.2 Requisits tècnics . . . . .	11
2.2 Presa de decisions . . . . .	11
2.2.1 Dispositiu electrònic . . . . .	12
2.2.2 Aplicació del terminal mòbil . . . . .	14
2.2.3 Protocol de comunicacions . . . . .	16
2.2.4 Aplicació de seguiment del tractament . . . . .	17
2.3 Escenari resultant . . . . .	19
<b>3 Disseny del protocol</b>	<b>21</b>
3.1 La tecnologia Bluetooth . . . . .	21
3.1.1 L'estàndard IEEE 802.15.1 . . . . .	21
3.1.2 Nocions sobre Bluetooth . . . . .	22
3.1.3 Topologia de xarxa . . . . .	26
3.1.4 Pila de procotols . . . . .	29
3.1.5 Perfils . . . . .	30
3.1.6 Establiment de connexió . . . . .	31
3.1.7 Seguretat . . . . .	33
3.1.8 Alternatives a Bluetooth . . . . .	35
3.2 Descripció del protocol . . . . .	38

3.2.1	Protocol dispositiu - terminal mòbil . . . . .	38
3.2.2	Protocol aplicació de seguiment - terminal mòbil . . . . .	43
<b>4</b>	<b>Aplicació de control</b>	<b>49</b>
4.1	Java 2 Micro Edition (J2ME) . . . . .	49
4.1.1	Introducció a J2ME . . . . .	50
4.1.2	Connected Limited Device Configuration (CLDC) . . . . .	52
4.1.3	Mobile Information Device Profile (MIDP) . . . . .	54
4.1.4	MIDlets . . . . .	56
4.1.5	Empaquetatge i distribució . . . . .	58
4.1.6	Record Management System (RMS) . . . . .	59
4.2	Paquets opcionals per a J2ME . . . . .	61
4.2.1	Java APIs for Bluetooth Wireless Technology (JABWT) . . . . .	61
4.2.2	Mobile Media API (MMAPI) . . . . .	65
4.2.3	Lightweight UI Toolkit (LWUIT) . . . . .	68
4.3	Desenvolupament de l'aplicació . . . . .	69
4.3.1	Entorn de desenvolupament . . . . .	70
4.3.2	Patró Model-Vista-Controlador (MVC) . . . . .	70
4.3.3	Estructura i funcionament . . . . .	71
<b>5</b>	<b>Dispositiu electrònic</b>	<b>77</b>
5.1	Disseny de la Unitat de Control (UC) . . . . .	77
5.1.1	Comandes SPP-AT d'Infineon . . . . .	77
5.1.2	Diagrama d'estats . . . . .	78
5.2	Desenvolupament del simulador . . . . .	79
5.2.1	Java Communications API (COMMAPI) . . . . .	80
5.2.2	Development Kit per a Infineon PBA31308/2 . . . . .	81
5.2.3	Estructura i funcionament . . . . .	82
5.2.4	Traces de la UC . . . . .	83
5.3	Preparació del dispositiu . . . . .	87
5.3.1	Estructura i components . . . . .	87
5.3.2	Programació del microcontrolador Infineon XC864 . . . . .	89
5.3.3	Sensibilitat del sensor . . . . .	90
5.3.4	Balanç energètic . . . . .	92
5.3.5	Prototip final . . . . .	94
<b>6</b>	<b>Banc de proves</b>	<b>95</b>
6.1	Test i proves pilot . . . . .	95
6.2	Temps de resposta d'alarma . . . . .	96
6.3	Coexistència amb altres dispositius . . . . .	100



<b>7</b>	<b>Aplicació de seguiment</b>	<b>103</b>
7.1	Java 2 Standard Edition (J2SE) . . . . .	103
7.1.1	Introducció a J2SE . . . . .	103
7.1.2	JavaDB/Derby technology database . . . . .	104
7.2	Paquets opcionals per a J2SE . . . . .	106
7.2.1	BlueCove API . . . . .	106
7.2.2	JFreeChart/JCommon API . . . . .	107
7.2.3	iText PDF API . . . . .	108
7.3	Desenvolupament de l'aplicació . . . . .	108
7.3.1	Entorn de desenvolupament . . . . .	108
7.3.2	Disseny de la Base de Dades de JavaDB . . . . .	109
7.3.3	Implementació del patró MVC . . . . .	111
7.3.4	Estructura i aparença . . . . .	111
<b>8</b>	<b>Preparació comercial</b>	<b>115</b>
8.1	Plataforma de descàrrega per SMS . . . . .	115
8.1.1	WAP i la tecnologia WAP-Push . . . . .	115
8.1.2	Posada en funcionament . . . . .	116
8.2	Web corporatiu de PPStop . . . . .	119
8.3	Signatura del MIDlet . . . . .	120
8.3.1	Problemàtica . . . . .	120
8.3.2	Signar un MIDlet . . . . .	121
<b>9</b>	<b>Conclusions i línies futures</b>	<b>125</b>
9.1	Conclusions . . . . .	125
9.2	Línies futures . . . . .	127
	<b>Apèndix</b>	<b>129</b>
<b>A</b>	<b>Simulador del dispositiu electrònic</b>	<b>129</b>
A.1	Diagrama de classes del simulador . . . . .	131
A.2	JavaDoc del simulador . . . . .	133
A.3	Codi del simulador . . . . .	176
<b>B</b>	<b>Dispositiu electrònic</b>	<b>199</b>
B.1	Transmissor Bluetooth Infineon eUniStone PBA31308/2 SPP-AT . . . . .	201
B.2	Microcontrolador Infineon XC864 8-bit Single Chip . . . . .	204
B.3	Bateria EEMB CR1/3N . . . . .	207
B.4	Comandes/Respostes SPP-AT d'Infineon . . . . .	209
B.5	Codi del microcontrolador . . . . .	218
B.6	Esquema de connexions . . . . .	239
B.7	Layout, components i encapsulat . . . . .	241

<b>C</b>	<b>Aplicació de control</b>	<b>243</b>
C.1	Diagrama de classes de l'aplicació . . . . .	245
C.2	JavaDoc de l'aplicació . . . . .	247
C.3	Codi de l'aplicació . . . . .	369
C.4	Guia de l'Usuari . . . . .	415
C.5	Terminals mòbils compatibles . . . . .	428
<b>D</b>	<b>Aplicació de seguiment</b>	<b>433</b>
D.1	Diagrama de classes de l'aplicació . . . . .	435
D.2	JavaDoc de l'aplicació . . . . .	437
D.3	Codi de l'aplicació . . . . .	534
D.4	Guia de l'Usuari . . . . .	587
D.5	Exemple d'informe . . . . .	599
	<b>Acrònims</b>	<b>605</b>
	<b>Bibliografia</b>	<b>610</b>

# Llista de figures

1.1	Aparells alàmbrics per al tractament de l'enuresi . . . . .	3
1.2	Planificació global del projecte . . . . .	4
1.3	Diagrama de planificació Gantt del projecte PPStop . . . . .	7
2.1	Escenari resultant sense coexistència amb altres dispositius . . . . .	19
3.1	Banda de freqüència de Bluetooth. Coexistència amb 802.11b/g . . . . .	23
3.2	Estructura d'un paquet Bluetooth . . . . .	25
3.3	Topologia de Bluetooth . . . . .	28
3.4	Pila de protocols Bluetooth . . . . .	30
3.5	Perfils de Bluetooth definits pel SIG . . . . .	31
3.6	Modes de funcionament per a la creació d'una connexió Bluetooth . . . . .	33
3.7	Diagrama de flux general del sistema . . . . .	39
3.13	Protocol general de comunicació entre aplicació de seguiment i terminal . . . . .	44
3.8	Protocol general de comunicació entre dispositiu electrònic i terminal . . . . .	45
3.9	Cas concret del protocol: terminal sense servei PPStop . . . . .	46
3.10	Cas concret del protocol: terminal amb servei PPStop no correspost . . . . .	46
3.11	Cas concret del protocol: connexió satisfactòria a 2 terminals . . . . .	47
3.12	Cas concret del protocol: connexió KO per al 1r terminal i OK per al 2n . . . . .	47
4.1	Edicions de la família Java . . . . .	50
4.2	Plataforma Java 2 Micro Edition . . . . .	51
4.3	Configuració CLDC . . . . .	52
4.4	Estats d'un MIDlet . . . . .	57
4.5	Empaquetatge d'un MIDlet . . . . .	58
4.6	Distribució d'un MIDlet . . . . .	59
4.7	Estructura de la Record Management System . . . . .	60
4.8	Resum de JABWT . . . . .	62
4.9	Elements d'un MIDlet Bluetooth amb JABWT . . . . .	65
4.10	Arquitectura i classes de MMAPI . . . . .	66
4.11	Estats d'un objecte de la classe Player . . . . .	67
4.12	Jerarquia de components de LWUIT simplificada . . . . .	69
4.13	Patró Model-Vista-Controlador . . . . .	71
4.14	Estructura simplificada de l'aplicació de control . . . . .	75
4.15	Captures de l'aplicació PPStop . . . . .	76

5.1	Diagrama d'estats de la UC . . . . .	80
5.2	Visió global de la Java COMMAPI . . . . .	81
5.3	Development Kit per a Infineon PBA31308/2 . . . . .	82
5.4	Estructura simplificada del simulador . . . . .	83
5.5	Captures del simulador . . . . .	83
5.6	Escenaris per al testeig de la UC mitjançant el simulador . . . . .	84
5.7	Estructura i elements del dispositiu electrònic . . . . .	87
5.8	Estructura interna del XC864 . . . . .	88
5.9	Components del dispositiu electrònic . . . . .	88
5.10	Easy Kit for Developing per a Infineon XC864 . . . . .	89
5.11	Circuit comparador per definir el nivell de sensibilitat del sensor d'humitat . . . . .	90
5.12	Primer prototip PPStop . . . . .	94
6.1	Pack de prototips per a les proves pilot . . . . .	96
6.2	Gràfica del temps d'alarma en funció de la distància . . . . .	98
6.3	Escenari global real amb possible coexistència amb altres dispositius PPStop . . . . .	100
7.1	Java 2 Platform Standard Edition 5 . . . . .	105
7.2	Utilització de JavaDB per a persistència de dades . . . . .	106
7.3	Llibreria BlueCove per a Bluetooth . . . . .	107
7.4	Disseny de la base de dades de l'aplicació . . . . .	109
7.5	Estructura simplificada de l'aplicació de seguiment . . . . .	113
7.6	Captura de la pantalla principal de l'aplicació de seguiment . . . . .	114
7.7	Captura d'un informe PDF generat per l'aplicació . . . . .	114
8.1	Plataforma de descàrrega WAP-Push per a PPStop . . . . .	117
8.2	Estructura de la web . . . . .	120
8.4	Signatura del MIDlet . . . . .	122
8.3	Captures de la pàgina web corporativa de PPStop . . . . .	124

# Llista de taules

3.1	Potències i abasts per a les diferents classes de dispositius Bluetooth . . .	23
3.2	Banda de freqüències i canals Bluetooth segons país o zona . . . . .	24
3.3	Característiques tècniques dels paquets Bluetooth . . . . .	27
3.4	Diferència entre els diferents sistemes de comunicació inalàmbrica . . . .	37
3.5	Dades intercanviables dins del protocol DE-TM . . . . .	42
3.6	Dades intercanviables dins del protocol AS-TM . . . . .	43
4.1	Paquets inclosos en la configuració CLDC . . . . .	53
4.2	Diferències entre els paquets MIDP 1.0 i MIDP 2.0 . . . . .	56
4.3	Paquets de classes de l'aplicació del terminal mòbil . . . . .	72
5.1	Descripció de les comandes AT utilitzades . . . . .	78
5.2	Descripció de possibles respostes a les comandes AT . . . . .	79
5.3	Traces de la UC obtingudes amb el simulador . . . . .	85
5.4	Valor dels pins en funció del nivell de sensibilitat del sensor d'humitat . .	92
5.5	Consum del dispositiu electrònic segons el seu estat . . . . .	93
6.1	Mostres de temps(ms) d'alarma segons distància, per a 1 i 2 terminals . .	97
6.2	Mitjanes de temps(ms) d'alarma segons distància, per a 1 i 2 terminals .	98
7.1	Paquets de Java Platform Standard Edition . . . . .	104
7.2	Restriccions i descripció dels atributs de la taula "Pacients" . . . . .	110
7.3	Restriccions i descripció dels atributs de la taula "Alarmes" . . . . .	110
7.4	Paquets de classes de l'aplicació de seguiment . . . . .	112



# Llista de codis

7.1	Codi SQL de creació de taules per a JavaDB . . . . .	110
8.1	(SMSServer) SMSServer.py . . . . .	117
8.2	(SMSServer) PPStop.py . . . . .	119
A.1	(Simulador) ComunicacioSerie.java . . . . .	176
A.2	(Simulador) Dispositiu.java . . . . .	177
A.3	(Simulador) GUI.java . . . . .	178
A.4	(Simulador) Model.java . . . . .	185
A.5	(Simulador) Simulador.java . . . . .	190
A.6	(Simulador) UnitatControl.java . . . . .	191
A.7	(Simulador) Vista.java . . . . .	197
B.1	(Microcontrolador) IO.c . . . . .	218
B.2	(Microcontrolador) IO.h . . . . .	219
B.3	(Microcontrolador) MAIN.c . . . . .	220
B.4	(Microcontrolador) MAIN.h . . . . .	227
B.5	(Microcontrolador) PS.c . . . . .	233
B.6	(Microcontrolador) PS.h . . . . .	234
B.7	(Microcontrolador) T2.c . . . . .	234
B.8	(Microcontrolador) T2.h . . . . .	235
B.9	(Microcontrolador) UART.c . . . . .	235
B.10	(Microcontrolador) UART.h . . . . .	237
C.1	(AppControl) Alarma.java . . . . .	369
C.2	(AppControl) Audio.java . . . . .	369
C.3	(AppControl) Connexio.java . . . . .	371
C.4	(AppControl) Servei.java . . . . .	372
C.5	(AppControl) CONST.java . . . . .	375
C.6	(AppControl) LANG.java . . . . .	377
C.7	(AppControl) BD.java . . . . .	381
C.8	(AppControl) Config.java . . . . .	382
C.9	(AppControl) Historial.java . . . . .	386
C.10	(AppControl) Model.java . . . . .	387
C.11	(AppControl) RMS.java . . . . .	393
C.12	(AppControl) PPStop.java . . . . .	394
C.13	(AppControl) Utils.java . . . . .	396
C.14	(AppControl) UtilsVista.java . . . . .	398

C.15 (AppControl) FormAlarma.java . . . . .	400
C.16 (AppControl) FormConfig.java . . . . .	401
C.17 (AppControl) FormHistorial.java . . . . .	403
C.18 (AppControl) FormMenu.java . . . . .	405
C.19 (AppControl) FormResum.java . . . . .	406
C.20 (AppControl) FormSplash.java . . . . .	408
C.21 (AppControl) Vista.java . . . . .	408
D.1 (AppSeguiment) Bluetooth.java . . . . .	534
D.2 (AppSeguiment) Connexio.java . . . . .	536
D.3 (AppSeguiment) CONST.java . . . . .	537
D.4 (AppSeguiment) LANG.java . . . . .	538
D.5 (AppSeguiment) Alarma.java . . . . .	541
D.6 (AppSeguiment) DB.java . . . . .	542
D.7 (AppSeguiment) Historial.java . . . . .	546
D.8 (AppSeguiment) Model.java . . . . .	548
D.9 (AppSeguiment) Pacient.java . . . . .	552
D.10 (AppSeguiment) App.java . . . . .	554
D.11 (AppSeguiment) Grafics.java . . . . .	555
D.12 (AppSeguiment) Informe.java . . . . .	558
D.13 (AppSeguiment) Utils.java . . . . .	565
D.14 (AppSeguiment) UtilsVista.java . . . . .	567
D.15 (AppSeguiment) DialogPacient.java . . . . .	568
D.16 (AppSeguiment) GUI.java . . . . .	573
D.17 (AppSeguiment) SplashScreen.java . . . . .	584
D.18 (AppSeguiment) Vista.java . . . . .	585



# Capítol 1

## Introducció, motivació i objectius

### 1.1 L'enuresi i el seu tractament

L'enuresi[1] és el terme mèdic utilitzat per definir un trastorn present en nens que consisteix en la presència de miccions d'orina incontrolades —siguin diürnes, nocturnes o mixtes— més enllà de l'edat en la que el nen hauria d'arribar a adquirir el control vesical. Afecta aproximadament a un 10% de la població infantil i amb major prevalença en els nois.

Es poden trobar tres tipus d'enuresi: nocturna, diürna o mixta. Els nens amb enuresi nocturna mullen el llit mentre estan dormint i no són capaços de despertar-se. Si el nen no presenta cap símptoma urinari durant el dia, se la denomina Enuresi Nocturna Monosintomàtica Primària (ENMP). Per la seva forma de presentació també es pot classificar per enuresi primària quan el nen mai ha presentat continència urinària durant un període perllongat, o per enuresi secundària quan el nen mulla la roba durant el dia o la nit després d'un període de continència de 6 a 12 mesos. La denominació d'enuresi diürna primària fa referència a si persisteix durant el dia després dels 4 anys i enuresi nocturna primària a si persisteix durant la nit després dels 6 anys. De tots els tipus esmentats, l'enuresi nocturna o *ENMP* n'és el tipus més freqüent i constitueix el 85% del total dels casos.

Segons cita en un article el Dr. Santiago Garcia-Tornel, cap de la Unitat de Trastorns d'esfínters de l'*Hospital Sant Joan de Déu de Barcelona* —unitat especialitzada en enuresi— es tracta d'un trastorn que *"no és psicològic primari, no es cura sol, no és normal després dels 7 anys, no es produeix per beure excessivament i, sobretot, és tractable"*[2]. Habitualment no s'indica tractament abans dels 6 anys perquè la taxa de curació espontània és elevada. Fins i tot després dels 6 anys, la curació espontània en l'enuresi primària ascendeix al 15%. La resta de casos però, tal i com afirma el Dr. Santiago García-Tornel, cal que siguin tractats.

Els estudis científics i psicològics indiquen que l'enuresi pot tenir tant un tractament

actiu com passiu. El tractament actiu està basat en fàrmacs sintètics com la *desmopresina*, mentre que el tractament passiu es basa en el sistema de l'alarma sonora[3] —cas que es tracta en aquest projecte— i és el mètode més utilitzat en els casos d'enuresi nocturna o *ENMP*.

Les alarmes d'enuresi ensenyen al nen a despertar-se davant la sensació de bufeta plena. Un sistema d'alarma d'enuresi tradicional és un aparell electrònic provist d'un sensor d'humitat i connectat a un timbre sonor que es dispara automàticament quan el sensor entra en contacte amb la humitat. Té l'objectiu d'actuar com a estímul; pretén despertar al pacient en el moment de la micció per a què progressivament vagi aprenent a reconèixer el buidat imminent de la bufeta i tingui temps d'anar al lavabo. Aquest tipus d'alarmes s'utilitzen en nens a partir dels 7 anys i amb aquest mètode de tractament s'ha observat una taxa de millora del 70%.

## 1.2 Motivació

En els darrers anys la tecnologia ha anat adquirint un gran protagonisme dins de les nostres vides. Un dels casos més espectaculars és el dels telèfons mòbils, que s'han convertit en un instrument imprescindible en la vida quotidiana. Això ho demostra el fet que avui en dia és molt difícil trobar una persona que no disposi d'un terminal mòbil. Tal n'ha sigut l'acceptació que cada dia apareixen nous models que incorporen més funcionalitats i més suport a noves tecnologies que els seus predecessors, de manera que el que en un principi havia de ser un instrument per a realitzar únicament trucades telefòniques avui en dia s'ha convertit en molt més que això. És d'allò més normal trobar un terminal mòbil que incorpori per exemple una càmera de fotos digital, una video-càmera per a fer videoconferència, un receptor de radio o fins i tot un reproductor d'arxius *mp3*.

Un dels trets característics més importants dels terminals d'avui en dia és la gran quantitat d'informació que han de gestionar. Fotografies, arxius d'àudio, programes, agendes... la taxa d'informació present en un terminal mòbil creix exponencialment en el temps, i és per això que d'uns anys cap aquí aquests terminals s'han provist de noves tecnologies que els permeten la connectivitat amb altres dispositius per tal de compartir-la o transferir-la. Per exemple, la majoria de terminals mòbils presents en el mercat actual incorporen connectivitat Bluetooth o connectivitat Wireless Fidelity (Wi-Fi), tecnologies de transferència d'informació molt utilitzades en entorns de Personal Area Network (PAN) i de Local Area Network (LAN).

Tanmateix, l'univers multimèdia que ens envolta ha obligat a adaptar la telefonia mòbil a les necessitats actuals, incorporant en els terminals mòbils pantalles a color, capacitats de reproducció de contingut multimèdia —audio, video— i augmentant les capacitats de processament per a l'execució d'aplicacions específiques. Tot plegat ha esdevingut una autèntica revolució que ha convertit els primitius terminals mòbils de

## 1.2. MOTIVACIÓ

---

finals dels anys 90 en autèntiques màquines gairebé equiparables a un ordinador personal i amb capacitat per a dur a terme pràcticament qualsevol tasca. Estem al segle XXI, la filosofia de vida ha canviat totalment i a poc a poc cal anar adaptant les coses de tota la vida a aquest nou entorn que ens envolta.

En aquest sentit, es planteja la idea de projecte del departament d'I+D de l'*Hospital Sant Joan de Déu* de Barcelona. La idea consistia en adaptar a les noves tecnologies el sistema tradicional passiu de tractament de l'enuresi per a millorar les prestacions globals del sistema cablejat existent fins al moment, creant un nou sistema que fes us de les Tecnologies de la Informació i la Comunicació (TIC) i del qual se'n poden veure exemples en la figura 1.1. La proposta consisteix concretament en fer ús de la tecnologia Bluetooth[4] per desenvolupar un sistema d'alarma inalàmbic, substituint el timbre que alarma el pacient d'enuresi per una aplicació resident en el terminal mòbil del pacient que n'imités la seva funcionalitat.

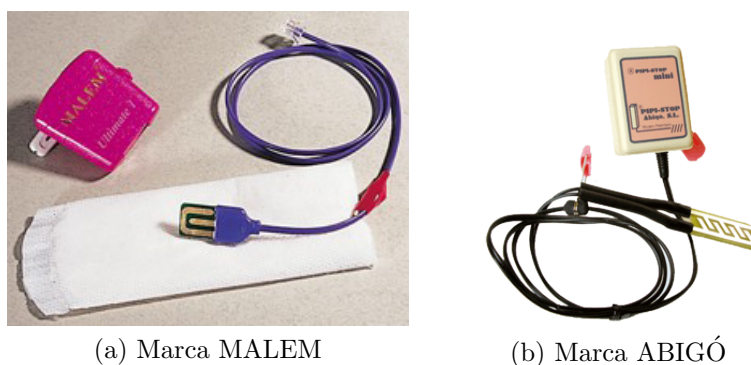


Figura 1.1: Aparells alàmbrics per al tractament de l'enuresi

No obstant, per a tirar endavant el projecte calia superar un repte inicial: calia determinar la viabilitat del sistema utilitzant la tecnologia Bluetooth. Hi ha molts factors que es van tenir en compte inicialment per creure que sí que era possible, com ara la possibilitat d'obtenció d'un temps de resposta d'alarma relativament curt mitjançant Bluetooth que no deteriorés l'eficiència del tractament, o com ara la viabilitat de desenvolupar un dispositiu electrònic Bluetooth que fos capaç de transmetre informació amb un consum energètic molt baix i que tingués un temps de vida suficientment llarg com per a poder ser comercialitzat.

Per tot això, des del Departament d'Enginyeria Telemàtica de la UPC i amb la confiança del departament d'I+D de l'*Hospital Sant Joan de Déu* es va decidir tirar endavant el projecte —projecte batejat amb el nom de *PPStop*, llegeixi's Pi-Pi-Stop— i sobre el qual se'n farà referència d'ara en endavant. Posteriorment, des del Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial també de la UPC es va donar suport per a complementar aquest projecte amb el desenvolupament d'una aplicació que donés suport de supervisió del tractament de l'enuresi en els pacients que utilitzen el sistema PPStop.

### 1.3 Objectius a assolir

Els objectius que es van definir per al present Projecte Final de Carrera eren els establerts en la primera fase del projecte PPStop, i part dels establerts per a la segona fase del projecte.

Aquesta primera fase englobava tot el procés de disseny i d'implementació del sistema inalàmbic Bluetooth, culminat amb l'obtenció d'un prototip que permetés dur a terme proves pilot del sistema per testejar-ne la seva eficàcia i analitzar-ne els resultats. El procés de disseny incloïa la presa de requisits del sistema a desenvolupar —tant funcionals com tècnics— i la presa de decisions sobre com havien de ser els diferents elements que intervenen en el sistema —dispositiu electrònic, aplicació del terminal mòbil, protocols de comunicació—. El procés d'implementació incloïa la programació de l'aplicació J2ME del terminal mòbil i la programació del dispositiu electrònic —unitat de control del microcontrolador— del dispositiu electrònic, prèvia simulació del mateix.

La segona fase englobava la correcció i millora de certs aspectes del sistema en funció dels resultats obtinguts en les proves pilot sobre el prototip obtingut en la primera fase, el desenvolupament de l'aplicació J2SE per a l'anàlisi i seguiment del tractament, i l'inici de la preparació comercial del sistema, és a dir, la implementació d'una plataforma de descàrrega de l'aplicació PPStop —a través de SMS— i la realització d'un web corporatiu que permetés difondre i promocionar tot el sistema PPStop desenvolupat.

No formaven part dels objectius a assolir el disseny i implementació de la placa electrònica del dispositiu electrònic. D'aquesta tasca se n'encarregava l'empresa *Electrònica Feixas*, tot i que ja es va preveure que hi hauria molts moments en que caldria treballar conjuntament, sobretot en la part de requisits del sistema i en la de programació del microcontrolador.

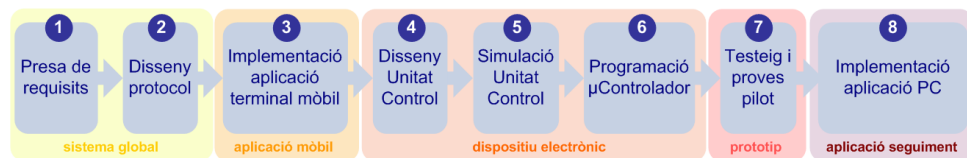


Figura 1.2: Planificació global del projecte

Per a assolir tots aquests objectius es va decidir seguir la planificació de treball de les tasques a realitzar per blocs que es mostra a la figura 1.2. Els objectius específics de cada bloc eren els que es descriuen tot seguit.

1. **Presa de requisits:** El·laborar d'una llista de requisits tècnics i funcionals que havia de complir el sistema, el més acurada possible que ajudés a tenir una visió global del sistema a dissenyar i implementar.
2. **Disseny del protocol:** Donats els requisits del sistema, dissenyar el protocol de comunicació entre el dispositiu electrònic i el terminal mòbil per a poder satisfer-los.

3. **Implementació de l'aplicació del terminal mòbil:** Implementar una aplicació J2ME per a terminals mòbils compatible amb el màxim nombre de models de terminal mòbil que fes ús de la tecnologia Bluetooth i permetés la interacció amb l'usuari, la correcta comunicació amb el dispositiu electrònic i l'aplicació de seguiment tot seguint el protocol dissenyat, i l'activació d'una alarma sonora.
4. **Disseny de la unitat de control:** Definir la màquina d'estats de la unitat de control del dispositiu electrònic que fent ús d'un determinat transmissor Bluetooth aconseguís seguir el protocol dissenyat en totes les seves possibilitats.
5. **Simulació de la unitat de control:** Implementar la unitat de control del dispositiu electrònic en un simulador que en permetés testejar el seu funcionament fent ús d'un mòdul de prova del transmissor Bluetooth, i que posteriorment fos capaç de ser exportada a codi d'un microcontrolador.
6. **Programació del microcontrolador:** Programar el microcontrolador a partir de la implementació de la unitat de control feta en el simulador.
7. **Testeig i proves pilot:** Obtenir un prototip que permetés testejar al laboratori tot el sistema en totes les seves possibilitats, verificar el compliment de tots els requisits definits en la presa de requisits i obtenir resultats concloents sobre el seu funcionament.
8. **Implementació de l'aplicació de seguiment del tractament:** Implementar una aplicació J2SE per a PC que fent ús de la tecnologia Bluetooth permetés l'adquisició del registre d'alarmes emmagatzemat per l'aplicació PPStop del terminal mòbil —tot seguint el protocol dissenyat— i el seu posterior tractament per mostrar diversos tipus d'informes —gràfics, PDF, etc— de seguiment del tractament a l'usuari final.

A banda d'una planificació global, també es va realitzar un diagrama de *Gantt* —mostrat en la figura 1.3 al final del capítol— amb una planificació més detallada de les tasques a realitzar, la duració prevista per a cadascuna d'elles i la dependència de les unes amb les altres. Tot plegat ha permès tenir en tot moment del projecte una referència de l'estat, les tasques i objectius complerts, les tasques i objectius pendents, etc., i en general ha permès una millor organització global. També va permetre detectar els possibles colls d'ampolla, i proposar solucions per mirar d'assolir els objectius proposats amb una millor eficiència.

## 1.4 Estructura de la memòria

A continuació es detallen les diferents parts que componen la memòria d'aquest projecte.

En el capítol 2 es citen els requisits que ha de complir el sistema i la presa de decisions que s'ha fet a l'hora de dissenyar els diferents elements que componen el sistema.

En el capítol 3 s'explica el funcionament de la tecnologia Bluetooth i les avantatges que aporta respecte altres sistemes inalàmbrics. En base a aquest funcionament, es detalla com s'ha dissenyat el protocol de comunicació entre els diferents elements del sistema i com queda descrit l'escenari resultant.

En el capítol 4, dins del marc de l'enginyeria del software, s'analitza el llenguatge de programació per a dispositius de capacitat limitada —*J2ME*— utilitzat per al desenvolupament de l'aplicació del terminal mòbil, els paquets addicionals que s'han hagut d'utilitzar sobre *J2ME* i com s'ha desenvolupat aquesta aplicació.

En el capítol 5, dins del marc de l'enginyeria electrònica, es detalla el disseny global del dispositiu electrònic encarregat de generar els avisos d'alarma als terminals mòbils i com s'ha dissenyat i simulat la unitat de control del microcontrolador que coordina les accions del transmissor Bluetooth del dispositiu.

En el capítol 6 s'analitzen certs resultats d'interès obtinguts mitjançant tests i una prova pilot realitzades a partir d'un primer prototipus del dispositiu electrònic. El capítol es centra majoritàriament en l'anàlisi del temps de generació d'alarma, punt més crític del sistema.

En el capítol 7, dins del marc de l'enginyeria del software, s'analitza breument el llenguatge de programació Java —edició J2SE— utilitzat per al desenvolupament de l'aplicació de seguiment del tractament, els paquets addicionals que s'han hagut d'utilitzar i com s'ha desenvolupat aquesta aplicació.

En el capítol 8 s'explica com s'han començat a preparar diferents elements no englobats en el disseny del sistema però necessaris per a una futura comercialització del producte obtingut. S'analitza la implementació de la plataforma de descàrrega per SMS, la creació del web corporatiu de PPStop i com cal preparar l'aplicació del terminal mòbil per a que sigui un producte de mercat.

Finalment, en el capítol 9 s'exposen les conclusions del projecte i les línies futures de treball sobre el sistema que s'ha desenvolupat.

Per completar tota la informació, s'han inclòs annexos que contenen tota la documentació necessària per a acabar de complementar les explicacions realitzades en els capítols anteriors. S'hi inclou els diagrames de classes, el javadoc i el codi sencer tant del simulador de la unitat de control com de l'aplicació del terminal mòbil i de l'aplicació de seguiment del tractament; l'esquema de connexions del dispositiu electrònic, tota la informació tècnica dels principals components electrònics que s'han fet servir i el codi del microcontrolador; i per acabar s'hi inclou també la guia d'usuari de l'aplicació del terminal mòbil i de l'aplicació de seguiment del tractament, i el llistat de models de terminals compatibles amb el sistema.

1.4. ESTRUCTURA DE LA MEMÒRIA

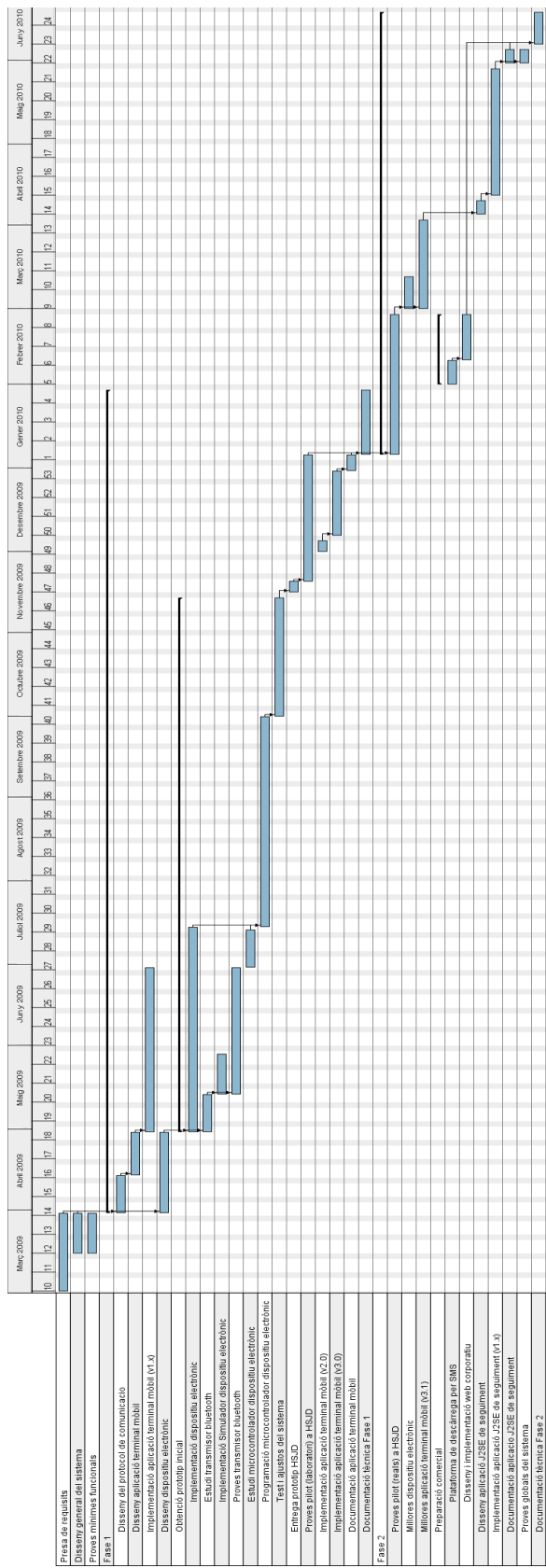


Figura 1.3: Diagrama de planificación Gantt del proyecto PPStop





# Capítol 2

## Disseny del sistema

### 2.1 Requisits del sistema

Per a dissenyar un sistema cal saber amb exactitud tot allò que ha fer i amb quines restriccions. És per això que previ al disseny del sistema s'ha procedit a la presa de requisits que ha de complir el projecte per a què sigui útil en el tractament de l'enuresi i aportí millores respecte el sistema d'alarma tradicional. A continuació es mostra una enumeració de requisits classificada segons es tracti de requisits funcionals —centrats en tot allò que el sistema ha de poder fer de cara a l'usuari final— o de requisits tècnics —centrats en aspectes no relacionats amb la manera de funcionar del sistema però que es veuen afectats per la tecnologia utilitzada—.

#### 2.1.1 Requisits funcionals

En referència al sistema en general, es tenen els següents requisits

1. Ha d'eliminar totalment el cablejat.
2. Ha de ser gestionat a través d'una aplicació en un terminal mòbil.
3. Ha de funcionar perfectament en un entorn domèstic.

En referència a l'aplicació del terminal mòbil, es tenen els següents requisits

4. Ha de poder ser descarregat mitjançant l'enviament d'un Short Message Service (SMS).
5. Ha de ser compatible amb el màxim nombre de terminals actualment disponibles al mercat.
6. Ha de tenir una interfície gràfica fàcil i intuïtiva.
7. Ha de poder-se definir sobre quin dispositiu electrònic ha de treballar.

8. Ha de ser utilitzable amb qualsevol dispositiu electrònic.
9. Ha de conèixer en tot moment l'estat de bateria i nivell de sensibilitat del dispositiu electrònic.
10. Ha de ser capaç de generar una alarma sonora quan així li ho indiqui el dispositiu electrònic.
11. Ha de ser capaç d'ajustar els paràmetres del dispositiu electrònic, com ara la sensibilitat del sensor, els terminals a alertar, o rotar l'ordre en que el dispositiu electrònic serveix alarmes als terminals mòbils.
12. Ha de guardar un historial d'alarmes generades.
13. Ha de tenir persistència de dades.
14. Ha de ser capaç de transferir l'historial d'alarmes a una aplicació externa.
15. Ha d'incorporar una guia d'usuari que en detalli el seu funcionament.
16. No ha d'interferir amb la resta de funcionalitats del terminal mòbil.

En referència al dispositiu electrònic, es tenen els següents requisits

17. Ha de generar una alarma en mullar-se el sensor d'humitat.
18. El sensor d'humitat no ha de saltar amb el contacte amb la pell del pacient.
19. Ha de ser capaç d'alertar més d'un terminal mòbil.
20. Ha de permetre de manera remota el canvi de sensibilitat del sensor d'humitat, iniciar una nova búsqueda de terminals i rotar l'ordre en que el dispositiu electrònic serveix les alarmes.
21. No ha de permetre el reemplaçament de la bateria
22. Ha de romandre en funcionament durant un temps pròxim a un any.
23. Ha de poder ser mullat i rentat.

En referència a l'aplicació de seguiment, es tenen els següents requisits

25. Ha de poder ser descarregat a través d'internet i ser fàcilment instal·lable.
26. Ha de ser multiplataforma.
27. Ha de tenir una interfície gràfica fàcil i intuïtiva.
28. Ha de poder gestionar diversos pacients.
29. Ha de ser capaç d'obtenir l'historial d'alarmes de l'aplicació del terminal mòbil.
30. Ha de tenir persistència de dades.
31. Ha de mostrar l'historial d'alarmes de cada pacient.
32. Ha de mostrar gràfiques estadístiques de les alarmes de cada pacient.
33. Ha de ser capaç de generar un informe en format PDF.
34. Ha d'incorporar una guia d'usuari que en detalli el seu funcionament.

### 2.1.2 Requisits tècnics

En referència al sistema en general, es tenen els següents requisits

1. Ha d'utilitzar la tecnologia Bluetooth en la comunicació.
2. Ha de vetllar en tot moment per la salut del pacient.
3. Ha de ser robust a la possibilitat de que els terminals mòbils a alertar puguin estar apagats o fora de rang de cobertura en un determinat moment.
4. Ha de ser robust a la coexistència amb altres sistemes *PPStop* dins d'un mateix rang de cobertura.
5. Ha de ser robust a la coexistència amb altres tecnologies inalàmbriques, sense percebre ni generar interferències.

En referència a l'aplicació del terminal mòbil, es tenen els següents requisits

6. Ha de ser lleuger per facilitar-ne la seva distribució.
7. La seva compatibilitat no ha de dependre del fabricant del terminal.

En referència al dispositiu electrònic, es tenen els següents requisits

8. Ha de tenir unes dimensions reduïdes.
9. La seva presència no ha de suposar una molèstia per al pacient.
10. Ha d'incorporar un encapsulat que mantingui la placa electrònica aïllada de la humitat en tot moment.
11. Ha de tenir consum energètic molt baix.
12. Ha de tenir un rang de cobertura equivalent almenys al d'un habitatge petit/mitjà.
13. El temps d'alarma per al terminal del pacient no ha de ser superior als 3 segons.
14. En cas de més d'un terminal, l'alarma no necessàriament ha de ser servida concurrentment.

En referència a l'aplicació de seguiment, es tenen els següents requisits

16. Ha de ser lleuger per facilitar-ne la seva distribució.
17. Ha d'incorporar un instal·lador que en faciliti la seva instal·lació.
18. Ha d'incorporar totes les dependències de llibreries externes.
19. La seva compatibilitat no ha de dependre de la plataforma sobre la qual s'executi.

## 2.2 Presa de decisions

A partir dels requisits esmentats en l'apartat 2.1 s'han pres una sèrie de decisions de disseny sobre el dispositiu electrònic, l'aplicació del terminal mòbil i la de seguiment, i

el protocol de comunicacions. Són el que conformen el disseny global del sistema, i és un pas bàsic per a poder assolir els objectius del projecte amb garanties d'èxit. Tot seguit es comenten les decisions preses.

### 2.2.1 Dispositiu electrònic

El dispositiu electrònic es pot considerar l'element central del sistema. És el dispositiu encarregat de generar els avisos d'alarma al terminal mòbil i, per tant, és clau en el disseny del sistema. El seu disseny condiciona el disseny de la resta dels elements.

A nivell funcional, sabem que haurà de generar una alarma quan el nen s'orini, per tant ha d'incloure un sensor d'humitat que pugui detectar aquest fet i generi un event que pugui ser reconegut pel dispositiu i actuar en conseqüència. Cal dotar el dispositiu d'algun tipus d'intel·ligència que sigui capaç de percebre events i de dur a terme un seguit d'accions, i per a això el dispositiu ha d'incorporar un microcontrolador. Els microcontroladors permeten executar codi —una unitat de control per exemple—, activar ports, modificar valors de registres, percebre events en base a interrupcions, emmagatzemar dades en memòria, etc. Així doncs sembla clar que n'és imprescindible la seva utilització. No obstant, calen més elements. S'ha parlat de la necessitat de generar un avís d'alarma a un terminal mòbil, i aquest avís ha de ser generat mitjançant tecnologia Bluetooth[5]. Per tant, s'ha d'incorporar algun component electrònic que permeti realitzar una búsqueda de terminals bluetooth en un determinat rang de cobertura, i permeti també la realització d'una connexió bluetooth a un determinat dispositiu per tal de dur a terme un intercanvi de dades. Per a aquesta finalitat existeixen transmissors Bluetooth que n'implementen la seva pila de protocols i permeten realitzar aquestes accions a base d'enviar-los una sèrie d'ordres —comandes—, per tant caldrà utilitzar-ne un. Es podria dir que aquests elements —sensor d'humitat, microcontrolador i transmissor Bluetooth— són els elements principals que ha de tenir el dispositiu electrònic. No obstant, cal tenir en compte també l'alimentació del sistema. S'ha de dotar el dispositiu d'alguna font d'energia que permeti el correcte funcionament d'aquests elements i tingui suficient autonomia com per fer que el dispositiu electrònic tingui un temps de vida d'aproximadament 1 any.

Tenint en compte tot això i que a més a més el dispositiu ha de tenir dimensions el més reduïdes possible, s'han escollit components en el mercat que satisfacin les premises esmentades i que s'adaptin a les necessitats i requisits del sistema. Per al sensor d'humitat s'utilitzarà un sensor convencional de doble pista en forma de "tira", que amb la conducció entre les seves pistes mitjançant l'orina provoqui un pas de corrent a través seu i, per tant, una caiguda de tensió. Per afinitat de la casa electrònica que dissenya el dispositiu electrònic —*Electrònica Feixas*— al fabricant alemany de microxips *Infineon*, s'han buscat aquests components dins de la seva gamma de productes. Com a microcontrolador, doncs, s'utilitzarà el model de 8 bits *Infineon XC864* —veure apèndix B.2—. Aquest microcontrolador té un consum relativament baix, té una capacitat de memòria flash de 4kB per a codi d'aplicació i té una memòria Read-Only Memory (ROM), Random Access Memory (RAM) i capacitat de processament suficients per a la funcionalitat que

se l'hi vol donar. A més a més permet ser "adormit" per a estalviar energia i ser "despertat" a través de la recepció d'un event. Com a transmissor Bluetooth s'ha decidit utilitzar el component *Infineon PBA31308/2 eUniStone* —veure apèndix B.1—. Aquest transmissor és de classe 2 (és a dir, té un rang de cobertura d'aproximadament 10m), funciona amb Bluetooth 2.0 + Enhanced Data Rate (EDR) i porta implementada tota la pila de protocols Bluetooth. Es controla mitjançant l'enviament de comandes AT específiques d'*Infineon* —veure apèndix B.4— i l'intercanvi de dades el fa a través de la Universal Asynchronous Receiver/Transmitter (UART). A més a més té implementat el Serial Port Profile (SPP) necessari per a l'intercanvi de dades de manera inalàmbrica a través de comunicació sèrie. No suporta més d'una connexió simultània —però això no suposa cap problema en el sistema *PPStop*— i té un consum raonable per ser un dispositiu d'aquestes característiques. No obstant, per estalviar energia i allargar el temps de vida del dispositiu electrònic caldrà que aquest treballi sempre com a màster, és a dir, serà ell el que iniciï totes les comunicacions en qualsevol cas. Això tindrà repercussions importants a l'hora de dissenyar el protocol de comunicació, i es veurà més endavant.

Pel que fa a la bateria, s'ha escollit una bateria de liti model CR1/3N —veure apèndix B.3— de la casa *EEMB*, fabricant especialitzat en bateries de liti. Aquesta ofereix una capacitat i un voltatge nominal aparentment suficients per a alimentar tots els components del dispositiu, té un límit de subministre de corrent prou elevat i a més a més té unes dimensions molt reduïdes.

Al marge de tots els components, caldrà dotar el dispositiu de la circuiteria necessària per definir el nivell de sensibilitat del sensor d'humitat i determinar l'estat de la bateria del dispositiu. Per al primer cas, s'utilitzarà un comparador amb una tensió de referència variable a través del microcontrolador en funció del nivell de sensibilitat del sensor. El segon cas es pot assolir utilitzant un nivell de tensió de referència invariable independentment de l'estat de la bateria que actuarà de patró i es compararia en l'AD del microcontrolador amb el valor de tensió real subministrat per la bateria. Aquest darrer cas es va decidir no implementar-se en la primera fase del Projecte i per tant no s'ha tractat en aquesta memòria.

Finalment, és necessari que el dispositiu electrònic sencer estigui perfectament encapsulat —tret, evidentment, del sensor d'humitat— per a evitar la penetració d'humitat en el seu interior i malmetre els components electrònics i tota la circuiteria. Per a aquesta finalitat, l'empresa d'enginyeria *IDOM* que actua com a coordinadora del projecte *PPStop* dissenyarà, fabricarà i proporcionarà un encapsulat perfectament adaptat a la forma del dispositiu per a poder aplicar-se en l'obtenció del prototip del dispositiu. Aquest encapsulat serà enresinat per tal d'aïllar hermèticament el seu contingut, i haurà de tenir una petita cavitat on introduir-hi la cinta de suport per a fixar la posició del dispositiu a la zona genital del pacient.

### 2.2.2 Aplicació del terminal mòbil

El terminal mòbil és l'altre element important del sistema, concretament l'aplicació que haurà d'executar per a poder gestionar el dispositiu electrònic i per a poder generar les alarmes sonores a partir de les connexions Bluetooth d'avís d'alarma rebudes per part del dispositiu electrònic. Per a la seva programació s'ha decidit utilitzar la plataforma de programació *J2ME*[6], l'edició del llenguatge *JAVA* per a dispositius amb capacitat gràfica i computacional reduïdes —i entre els quals es troben els terminals mòbils—. Actualment la gran majoria de terminals del mercat tenen suport per a executar aplicacions *J2ME* i el fet que treballi sobre una màquina virtual —l'encarregada d'interpretar el llenguatge java dels programes— n'extén la seva compatibilitat a pràcticament qualsevol tipus de terminal mòbil.

La plataforma *J2ME* permet també la incorporació de diverses Application Programming Interface (API) i llibreries opcionals per a afegir-hi suport per a diferents tipus de funcionalitat. Per exemple, per a treballar amb Bluetooth caldrà afegir les *APIs* de *JAVA* per a Bluetooth[7][8] —Java API for Bluetooth Wireless Technology (JABWT)—, que ofereixen una sèrie de classes i mètodes addicionals que permeten interactuar amb el transmissor Bluetooth del terminal mòbil. Això permetrà a l'aplicació crear un servei Bluetooth específic per a *PPStop* el qual pugui ser "descobert" pel dispositiu electrònic i al qual aquest s'hi pugui connectar, i tractar totes les connexions rebudes. Per a poder generar alarmes sonores serà necessari fer ús de la Mobile Media API (MMAPI)[9], que extén la funcionalitat de *J2ME* afegint-li suport per a àudio, vídeo i en general per a la reproducció de contingut multimèdia utilitzant els recursos multimèdia del terminal mòbil. Per a la programació de l'interfície gràfica d'usuari —Graphical User Interface (GUI)— s'utilitzarà l'*API* Lightweight User Interface Toolkit (LWUIT)[10] que permet desenvolupar interfícies gràfiques molt professionalitzades d'una manera relativament senzilla, malgrat que inicialment s'utilitzaran els components de *SWING*[11] incorporats en *J2ME* per a la creació d'una interfície gràfica simple però funcional que permeti implementar i testejar totes les funcionalitats requerides per a l'aplicació.

Amb totes les *APIs* necessàries ja incorporades, només caldrà parar atenció en programar correctament l'aplicació per a oferir a l'usuari les funcionalitats descrites en la presa de requisits. Per a la programació de l'aplicació s'utilitzarà un patró de programació molt utilitzat en enginyeria del software; el Model View Controller (MVC), que té com a objectiu principal separar les capes de presentació —vista—, tractament de dades —model— i control de l'aplicació i captació d'events de l'usuari —controlador—. La seva utilització permetrà estructurar bé l'aplicació a nivell de programació. A nivell de funcionalitat, s'ha decidit dividir el programa en els apartats que es detallen a continuació:

**Resum:** Apartat que s'encarregarà de mostrar tots els paràmetres del sistema i de l'aplicació. Haurà d'incorporar, com a mínim, l'estat de la bateria del dispositiu i el nivell de sensibilitat del sensor d'humitat.

**Buscar Terminals:** Apartat que permetrà activar el *flag* de búsqueda de terminals per

a que aquesta ordre pugui ser transmesa al dispositiu electrònic quan l'aplicació rebi una connexió Bluetooth des d'aquest.

**Modificar Sensibilitat:** Apartat que permetrà definir la nova sensibilitat del sensor d'humitat i activar el *flag* de canvi de sensibilitat per a que aquesta ordre pugui ser transmesa al dispositiu electrònic quan l'aplicació rebi una connexió Bluetooth des d'aquest.

**Configuració:** Apartat a través del qual es definiran les opcions de l'aplicació. Haurà d'incorporar, com a mínim, un camp d'introducció del *codi PPStop* del dispositiu electrònic al qual es vol pertànyer.

**Historial d'Alarmes:** Apartat que permetrà llistar i esborrar les alarmes enregistrades per l'aplicació.

**Rotar Terminals:** Apartat que permetrà activar el *flag* de rotació de terminals per a que aquesta ordre pugui ser transmesa al dispositiu electrònic quan l'aplicació rebi una connexió Bluetooth des d'aquest.

Pel que fa a la generació de l'alarma sonora, serà necessari fer ús d'un fitxer d'àudio el més lleuger possible per tal de que el seu procés de reproducció introdueixi el mínim retard possible en el temps d'alarma. Per a tal efecte s'utilitzarà el format Adaptive Multi Rate (AMR)[12], un format d'àudio d'ús molt extès en terminals mòbils i que ofereix nivells de qualitat prou acceptables per a *bitrates* molt baixos. Per a la conversió de fitxers d'àudio a aquest format caldrà fer ús d'una eina específica anomenada *NOKIA Multimedia Converter 2.0*[13] inclòs dins del paquet de *NOKIA Developers Tools* que ofereix aquest fabricant de telèfons mòbils.

Un altre aspecte a tenir en compte és que l'aplicació ha de ser capaç d'emmagatzemar informació de forma persistent en la memòria del telèfon mòbil, com és ara la configuració del programa i l'historial d'alarmes. L'emmagatzematge d'informació en terminals mòbils es realitza a la Record Management System (RMS), que actua com si es tractés d'una mena de base de dades amb la peculiaritat que aquestes han de ser "binaritzades" prèviament ja que *RMS* treballa a nivell de *byte*. El paquet *RMS* de *J2ME* ofereix suport per a emmagatzematge de dades a la *RMS* des de l'aplicació; caldrà doncs tenir en compte aquests aspectes.

Segons els requisits, l'aplicació ha de poder ser distribuïda mitjançant l'enviament d'un SMS per part de l'usuari. Per tant caldrà disposar d'una plataforma de descàrrega per SMS que sigui capaç de rebre SMSs, sigui capaç d'interpretar-los i pugui enviar SMSs de resposta indicant la localització del recurs —en aquest cas el fitxer *.jar* de l'aplicació— a la xarxa d'Internet per a que el terminal pugi descarregar-lo i procedir a la seva instal·lació. Una plataforma d'aquest tipus es pot implementar fent ús de la tecnologia *WAP* i *WAP-Push*[14], per exemple, per un sistema Unix si es té un mòdem General Packet Radio Service (GPRS) amb una tarja Subscriber Identity Module (SIM)

operativa existeix programari lliure per a tal objectiu. Aquest és el cas de *SMS Server Tools*[15], software molt complet en aquest sentit i que serà el que s'utilitzarà.

Finalment, cal que l'aplicació vagi acompanyada d'una guia de l'usuari que expliqui en detall el seu funcionament i totes les opcions disponibles en l'aplicació, i també cal elaborar una llista de terminals mòbils compatibles amb l'aplicació. Aquestes dos tasques també s'hauran de dur a terme durant el desenvolupament de l'aplicació.

### 2.2.3 Protocol de comunicacions

El protocol de comunicacions esdevé l'element d'interconnexió entre els diferents elements del sistema —dispositiu electrònic, aplicació de control del terminal mòbil i aplicació de seguiment—. El seu disseny ve clarament condicionat per 2 fets. El primer —explicat en l'apartat 2.2.1— és que el dispositiu electrònic sempre ha de ser el que iniciï les comunicacions i, conseqüentment, l'aplicació del terminal mòbil haurà d'estar sempre a l'escolta de connexions al servei Bluetooth que haurà creat específicament per a *PPStop*. El segon fa referència al requisit temporal de la generació de l'alarma, que no pot ser superior a 3s.

Bluetooth, tal i com es veurà en el capítol 3, té dos protocols —el Device Discovery Protocol (DDP) i el Service Discovery Protocol (SDP)— que utilitza per a buscar els terminals Bluetooth i els seus serveis dins del seu rang de cobertura. Per tal d'establir una connexió, el transmissor necessita saber els paràmetres del dispositiu de destí —adreça Media Access Control (MAC) i canal d'operació del servei—, cosa que implica que prèviament a la connexió ha d'utilitzar aquests dos protocols per obtenir aquests paràmetres. El més senzill i ideal fóra que cada vegada que el dispositiu electrònic hagués de generar un avís d'alarma utilitzés prèviament el *DDP* i el *SDP* i posteriorment establís la connexió, però això en el sistema *PPStop* no és possible ja que aquests protocols tenen una durada llarga i variable, de l'ordre dels 6 segons cadascun d'ells. L'alternativa viable per a *PPStop* passa per fer ús d'aquests protocols amb antelació a la generació de l'alarma, guardar-ne els resultats en memòria i utilitzar-los quan els necessiti per establir les connexions. Conseqüentment apareix el concepte de "terminal associat", que significa que un terminal dotat de l'aplicació de control *PPStop* ha estat "descobert" pel dispositiu electrònic i per tant l'ha inclòs a la llista de terminals que guarda a memòria de manera que podrà ser alarmat quan es generi un event procedent del sensor d'humitat. Tenint en compte aquesta limitació de disseny i juntament a la limitació comentada inicialment, caldrà dissenyar la resta del protocol per a donar suport a les funcionalitats requerides pel sistema.

A l'hora de generar els avisos d'alarma, el protocol haurà de tenir en compte que el transmissor Bluetooth que es farà servir en el dispositiu electrònic només podrà establir una connexió simultània i per tant, les connexions d'alarma hauran de ser generades seqüencialment. D'altra banda, una vegada establerta la comunicació el protocol haurà de fer ús del perfil *SPP* per a la transmissió de la informació ja que com s'ha comentat en



l'apartat 2.2.1 és el perfil que porta implementat el transmissor Bluetooth del dispositiu electrònic. El perfil *SPP* com es veurà en capítols posteriors, funciona sobre el protocol de transport Radio Frequency Communication (RFCOMM), que proporciona una emulació de ports sèrie i per tant permet la transmissió de dades binàries com si es tractés d'una comunicació convencional a través de *RS-232*. Caldrà també, doncs, establir un format determinat per a la transmissió d'informació en format binari en cada fase del protocol de manera que aquesta informació pugui ser correctament interpretada pel receptor. Pel que fa a la seguretat de la comunicació com que no es transfereix informació confidencial no caldrà establir cap mecanisme de seguretat que protegeixi les dades.

Tots aquests aspectes fan que les possibilitats de disseny del protocol quedin molt reduïdes. Això es veurà més detalladament en el capítol 3, que tot ell tracta del disseny del protocol.

### 2.2.4 Aplicació de seguiment del tractament

L'aplicació de seguiment és l'aplicació que s'encarregarà de recuperar els historials d'alarma dels terminals mòbils *PPStop* per tal de generar estadístiques, informes en format Portable Document Format (PDF), gràfiques d'evolució del tractament, etc, cosa que permetrà que un supervisor —típicament el doctor que tracti el pacient— pugui veure en el seu ordinador l'evolució del tractament del pacient. Com que es tracta d'una aplicació que ha d'interactuar amb l'aplicació del terminal mòbil, caldrà establir un protocol d'entesa entre ambdós que s'haurà de tenir en compte durant la implementació de l'aplicació del terminal mòbil.

Aquesta aplicació rebrà objectes d'una classe denominada *Historial* que emmagatzema totes les estampes de temps de les alarmes generades en el terminal mòbil. Com que el protocol utilitzat per a la transmissió d'informació, com hem dit, es basarà en el perfil *SPP*, caldrà "debinaritzar" aquest historial —prèviament "binaritzat" per l'aplicació del terminal mòbil— per a poder-lo interpretar en recepció. Aquest procés és conegut amb el nom de *deserialització* i com que *J2ME* no dóna suport a l'acció de *serialització* caldrà tenir en compte aquest aspecte i implementar manualment una funcions de *deserialització* compatible amb la funció de *serialització* programada en l'aplicació del terminal mòbil.

Per a la programació de l'aplicació s'ha decidit utilitzar la plataforma de programació *J2SE*[16], l'edició estàndar del llenguatge de programació *JAVA* ja que és una plataforma adequada per a la programació d'aplicacions que han de ser executades en *workstations* amb independència del seu sistema operatiu —aplicació multiplataforma—.

La plataforma *J2MS* permet també la incorporació de diverses API i llibreries opcionals per a afegir-hi suport per a diferents tipus de funcionalitat. Per exemple, per a treballar amb Bluetooth caldrà fer ús d'una API que implementi les especificacions de JSR-82, com per exemple *BlueCove*[17] o *Avetana*[18] —JABWT—, que ofereixen una sèrie de classes i mètodes addicionals que permeten interactuar amb el transmissor. Això permetrà a l'aplicació crear una connexió Bluetooth per a poder recuperar l'historial

d'alarmes. Per a poder emmagatzemar les dades de l'aplicació de forma persistent serà necessari utilitzar un sistema gestor de base de dades incrustat en l'aplicació com és el cas de *JavaDB*[19] i el corresponent driver per a permetre a l'aplicació l'accés a la base de dades. Per a la generació de gràfiques i informes en format *PDF* serà necessària la utilització de les *APIs* *JFreeChart*[20] i *iText PDF*[21] respectivament, que ofereixen una sèrie de classes de generació de diversos tipus de gràfiques i de generació i manipulació de documents *PDF*. Per a la programació de l'interfície gràfica d'usuari —GUI— s'utilitzaran els components de *SWING*[11] i Abstract Window Toolkit (AWT) incorporats en *J2SE* que permeten desenvolupar interfícies gràfiques molt professionalitzades d'una manera relativament senzilla.

Amb totes les *APIs* necessàries ja incorporades, només caldrà parar atenció en programar correctament l'aplicació per a oferir a l'usuari les funcionalitats descrites en la presa de requisits. Per a la programació de l'aplicació s'utilitzarà també el patró de programació MVC. A nivell de funcionalitat, s'ha decidit dividir el programa en els apartats que es detallen a continuació:

**Panell general:** Apartat principal de l'aplicació que permetrà a l'usuari seleccionar un pacient, mostrar el seu historial d'alarmes i les gràfiques estadístiques, i desencadenar la resta d'accions sobre el pacient.

**Gestió de pacients:** Apartat que s'encarregarà donar d'alta un nou pacient, o modificar-ne/eliminar-ne un d'existent.

**Obtenir historial:** Apartat que permetrà obtenir l'historial d'alarmes d'un determinat pacient a través de Bluetooth.

**Generar informe:** Apartat que permetrà generar un informe de seguiment d'un determinat pacient en format *PDF*. Aquest informe haurà d'incorporar les dades del pacient, les gràfiques estadístiques de les seves alarmes i un llistat del seu historial d'alarmes.

Segons els requisits, l'aplicació ha de poder ser descarregada a través d'internet i ser fàcilment instal·lable en qualsevol plataforma. Per tant caldrà generar un instal·lador de l'aplicació que sigui multiplataforma i que sigui força lleuger per a facilitar la distribució de l'aplicació. Un instal·lador d'aplicacions *JAVA* que compleix aquests requisits és l'*InstallBuilder*[22] de *BitRock*, software molt complet en aquest sentit i que serà el que s'utilitzarà.

Finalment, cal que l'aplicació vagi acompanyada d'una guia de l'usuari que expliqui en detall el seu funcionament i totes les opcions disponibles en l'aplicació.

### 2.3 Escenari resultant

Finalment, una vegada dissenyat el sistema, es pot representar l'escenari resultant de combinar tots els elements tal i com es mostra en la figura 2.1. Aquest escenari no és real ja que no il·lustra la coexistència del sistema amb altres dispositius i terminals *PPStop* dins d'un mateix rang de cobertura malgrat sí que s'ha tingut en compte aquest aspecte durant la fase de disseny. Aquest tema es tractarà amb més detall en l'apartat 6.3.

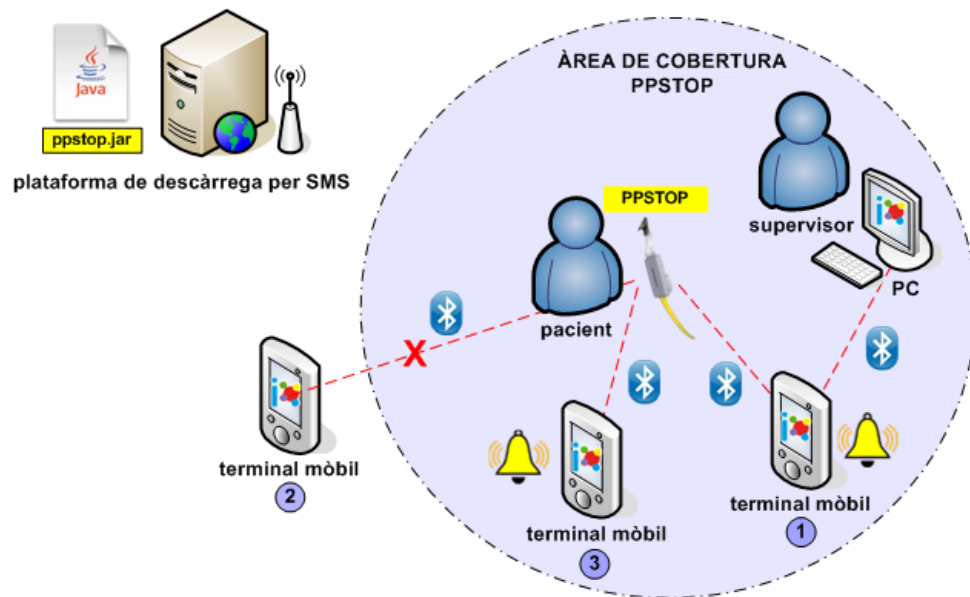


Figura 2.1: Escenari resultant sense coexistència amb altres dispositius



# Capítol 3

## Disseny del protocol

### 3.1 La tecnologia Bluetooth

Bluetooth[23] és una tecnologia de comunicació inalàmbrica omnidireccional ideada per a dispositius de baix consum i per a comunicacions en un entorn de Wireless Personal Area Network (WPAN), caracteritzada per ser xarxes de curta distància i habitualment d'àmbit personal. Aquesta tecnologia s'aplica a la comunicació sense cables entre dispositius portàtils com ara telèfons mòbils, PDAs, Pocket PCs o fins i tot perifèrics d'ordinador com ratolins o teclats.

La tecnologia va sorgir amb la idea d'intercomunicar dispositius mòbils mitjançant una interfície de baix cost i consum, ja que la majoria d'ells s'alimentaven a través d'una bateria. Així va ser com l'any 1994, Ericsson va proposar la tecnologia que en donava solució: la tecnologia Bluetooth.

La idea ràpidament va agradar a altres companyies com *IBM*, *Intel*, *Toshiba* i *Nokia*. L'any 1998 aquestes, juntament amb Ericsson van crear el Special Interest Group (SIG)[4] de Bluetooth amb la finalitat d'establir un estàndard de la tecnologia. A mesura que la tecnologia es va anar popularitzant el grup va anar creixent amb l'entrada d'empreses com *Microsoft*, *Motorola*, *3Com* o *Lucent* fins a arribar a les més de deu mil que actualment hi estan agregades i de sectors tan variats com les telecomunicacions, la informàtica, l'automobilisme, la música, l'automatització industrial o el tèxtil.

Actualment una dels majors àmbits d'ús de Bluetooth és sense cap dubte els telèfons mòbils i la gran majoria de terminals actualment en el mercat ja incorporen suport per a aquesta tecnologia.

#### 3.1.1 L'estàndard IEEE 802.15.1

No va ser fins l'any 2002 que l'Institute of Electrical and Electronics Engineers (IEEE) va publicar l'estàndard *IEEE 802.15.1*[24][5] on es defineixen les característiques de les co-

municacions Bluetooth amb l'objectiu d'assegurar la interoperabilitat entre els dispositius fabricats per les diferents companyies.

L'estàndard defineix Bluetooth com un sistema de comunicacions inalàmbric amb el qual es permet transmetre veu i dades a través d'un enllaç de radiofreqüència. Les principals propietats o característiques de l'estàndard són:

- a) Facilita la comunicació entre dispositius portàtils i/o dispositius fixes o estàtics, eliminant el cablejat a l'hora d'establir la comunicació entre ells.
- b) És un sistema de baix cost i consum, cosa que el fa apte per a dispositius alimentats per bateries.
- c) Ofereix la possibilitat de crear xarxes petites i de curt abast —anomenades *pi-conets*— entre els dispositius que utilitzen la mateixa tecnologia. Aquestes xarxes alhora poden unir-se entre si formant-ne així de més extenses —anomenades *scat-ternets*—.

La tecnologia Bluetooth defineix tant les especificacions de software com les de hardware que assegurin la compatibilitat entre fabricants. Utilitza la modulació Gaussian Frequency Shift Keying (GFSK) per a modular les dades a una freqüència pròxima als 2.4GHz en la banda de freqüència no regulades Industrial Scientific and Medical (ISM). Amb el pas dels anys, han anat apareixent versions de l'estàndard que aportaven millores generalment en aspectes referents a capacitat de transmissió i immunitat a interferències. La versió més recent és la Bluetooth 3.0 apareguda a mitjans del passat 2009 i que, combinada amb la tecnologia *Wi-Fi* ja ofereix taxes de transferència teòriques de fins a 24Mbps. No obstant, les versions més esteses actualment en el mercat són la 2.0 i la 2.1 acompanyades de EDR, que optimitzen considerablement el consum de potència i ofereixen una capacitat de transmissió de dades teòrica de fins a 3Mbps.

### 3.1.2 Nocions sobre Bluetooth

En aquest apartat es pretén analitzar les principals característiques i conceptes que defineixen l'estàndard Bluetooth.

#### Banda d'operació

Bluetooth, com s'ha comentat anteriorment, opera a la banda radio *ISM* a compresa entre els 2400MHz i els 2483.5Mhz, on coexisteix amb la tecnologia inalàmbrica *Wi-Fi* —estàndard 802.11 de la *IEEE*—. En països com Espanya, França o Japó l'ús de Bluetooth està limitat a un sub-rang freqüencial (veure taula 3.2). En el cas concret d'Espanya, per exemple, aquesta va dels 2445MHz fins els 2475Mhz.

### 3.1. LA TECNOLOGIA BLUETOOTH

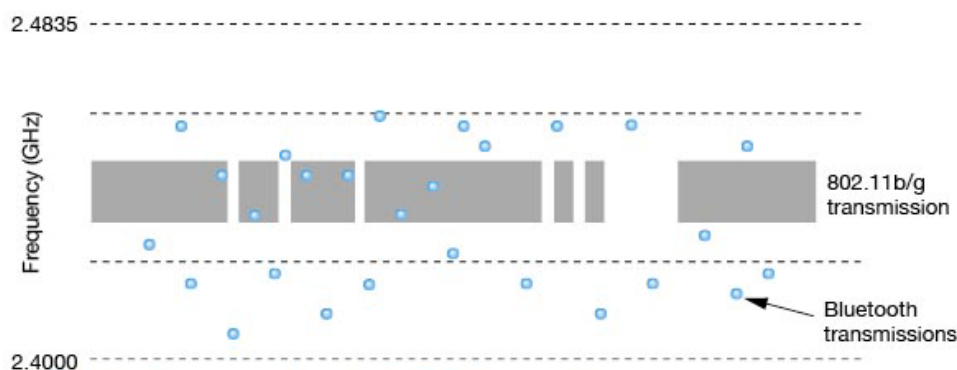


Figura 3.1: Banda de freqüència de Bluetooth. Coexistència amb 802.11b/g

#### Potència consumida i abast

L'abast —o rang de cobertura— de Bluetooth depèn de la potència utilitzada en transmissió. Segons aquesta potència, els dispositius es divideixen en tres classes tal i com mostra la taula 3.1. Els de primera classe són els que emeten a major potència i, conseqüentment, tenen un abast superior que els de tercera classe que són els que consumeixen menys potència. La majoria dels dispositius Bluetooth existents en el mercat són de segona classe; transmeten amb una potència nominal de sortida de 0dBm i una potència màxima de 4dBm i tenen un rang de cobertura d'aproximadament 10 metres en un escenari lliure d'obstacles. Aquest darrer és el cas dels telèfons mòbils, on s'ha aconseguit que la potència consumida pel terminal sigui inferior al 3% de la potència total consumida per aquest.

Classe	Potència màxima (mW)	Potència màxima (dBm)	Abast aproximat (m)
Classe 1	100	20	100
Classe 2	2.5	4	10
Classe 3	1	0	1

Taula 3.1: Potències i abasts per a les diferents classes de dispositius Bluetooth

#### Salt de freqüència i immunitat a interferències

Degut a que la banda *IMS* és oberta a qualsevol tipus per ser una banda lliure, el sistema radio Bluetooth està preparat per a evitar interferències gràcies a un sistema que busca una part no utilitzada de l'espectre mitjançant un mecanisme de salt de freqüència anomenat Frequency Hopping Spread Spectrum (FHSS) i orientat a dispositius de baix consum com els telèfons mòbils.

La tècnica del salt de freqüència és aplicada a una altra velocitat i una curta longitud dels paquets (1600 salts/s), i divideix tot el rang freqüencial en diversos canals. Durant una connexió, els transceptors van "saltant" d'un canal a un altre pseudo-aleatòriament. D'aquesta manera, encara que en cada moment s'estigui utilitzant un ample de banda instantani petit, s'assegura que aquest canal estigui el més lliure d'interferències possible.

## Canal

Bluetooth utilitza un sistema *FH/TDD*, és a dir, utilitza la tècnica de salt de freqüència o Frequency Hop (FH) combinada amb la de Time Division Duplex (TDD). El canal queda dividit en intervals de  $625\mu\text{s}$  anomenats *slots* i a cadascun d'ells es produeix un salt de freqüència, donant lloc així a una freqüència de 1600 salts/s. Per aconseguir un esquema *TDD*, un paquet de dades ocupa un determinat *slot* per a l'emissió i un altre per a la recepció.

Dues o més unitats Bluetooth poden compartir el mateix canal dins d'una *piconet* —xarxa formada per diversos dispositius Bluetooth per a comunicar-se entre si—. Tal i com es veurà en l'apartat 3.1.3 d'aquest capítol, en aquestes xarxes només hi ha un únic dispositiu que actui de mestre controlant el tràfic de dades que generen les demés unitats que actuen com a esclaus, ja sigui enviant paquets al màster o rebent-los dels esclaus.

El salt de freqüència del canal està determinat per la seqüència del senyal —ordre en el qual arriben els salts— i per la seva fase. Amb Bluetooth, la seqüència queda fixada per la identitat de la unitat mestra de la *piconet* —un codi únic per a cada dispositiu— i per la seva freqüència de rellotge. Es per això que per a què una unitat esclava pugui unir-se a una *piconet* ha d'estar sincronitzada amb la freqüència del rellotge de la unitat mestra i, per tant, cal que faci un ajust en el seu propi rellotge que li permeti compartir la mateixa portadora de salt utilitzada pel mestre. Tal i com es mostra en la taula 3.2, en els països on la banda està oberta a 80 canals o més —separats cada 1MHz— hi ha definides 79 freqüències o salts de portadora i en aquells països on la banda és més estreta —cas d'Espanya per exemple— se n'han definit 23.

País o zona	Banda de freqüències (GHz)	Nº canals Bluetooth
<b>Espanya</b>	2.445 - 2.475	23
<b>Europa</b>	2.400 - 2.4835	79
<b>França</b>	2.4465 - 2.4835	23
<b>USA</b>	2.400 - 2.4835	79
<b>Japó</b>	2.471 - 2.497	23

Taula 3.2: Banda de freqüències i canals Bluetooth segons país o zona



#### Paquet Bluetooth

Amb Bluetooth totes les dades que s'envien a través del canal són fragmentades i enviades en paquets. Per tal de protegir la informació s'utilitzen codis detectors i/o correctors d'errors. Cada paquet està compost per un codi d'accés, una capçalera i un camp d'informació tal i com es veu a la figura 3.2 i descriu tot seguit:

- a) **Codi d'accés (*Access code*):** Està format per 72 bits, deriva de la identitat del màster de la *piconet* i serveix per a identificar els dispositius Bluetooth.
- b) **Capçalera (*Packet header*):** Té una longitud de 54 bits i conté informació de control; tres bits d'accés a adreça, tipus de paquet, bits de control de flux, bits per a la retransmissió automàtica i comprovació d'errors dels camps de la capçalera.
- c) **Camp de dades (*Payload*):** Camp de longitud compresa entre 0 i 2745 bits que conté la informació transmesa. Aquesta pot ser veu, dades, o fins i tot totes dues simultàniament.

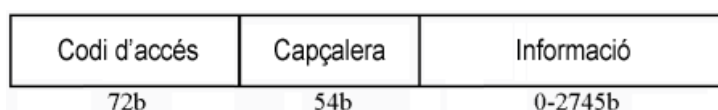


Figura 3.2: Estructura d'un paquet Bluetooth

Els paquets es poden classificar per tipus segons el nombre de *slots* (ranures) que ocupen i depenent de si els enllaços utilitzats són síncrons o asíncrons. Els paquets que ocupen 3 o 5 *slots* no utilitzen salts de freqüència i s'envien per la mateixa freqüència en tots els *slots* que ocupi el paquet. Una vegada finalitzada la transmissió del paquet es canvia la freqüència. Quan es rep un paquet en el terminal receptor, es compara el codi del paquet amb el codi d'accés i si no coincideixen aquest és considerat no vàlid en el canal i se n'ignora la resta del seu contingut.

#### Capacitat

La capacitat total del canal de comunicació és de 3Mbps però degut a l'*overhead* dels paquets transmesos —capçaleres i informació de control— la capacitat útil resultant és menor. A més a més, les capacitats de transmissió i recepció varien depenent del tipus d'enllaç establert entre els dispositius. Existeixen dos tipus principals d'enllaç:

- a) **Asynchronous Connection Less (ACL)** Enllaços asíncrons no orientats a connexió que permeten tant comunicacions simètriques com asimètriques i tant punt a punt com punt a multipunt. S'utilitzen per a la transmissió de dades i permeten aconseguir una taxa màxima de transferència de 723.2Kbps en transmissió i 57.6Kbps en recepció per a Bluetooth 1.1 i 1.2, i de 2.1Mbps per a Bluetooth 2.0 amb EDR 2.1.

- b) **Synchronous Connection Oriented (SCO)** Enllaços síncrons orientats a connexió que permeten comunicacions simètriques punt a punt i habitualment s'utilitzen en comunicacions de veu degut al seu caràcter síncron. Utilitza *slots* temporals iguals i consecutius tant en transmissió com en recepció amb l'objectiu d'evitar que l'usuari percebi retards en la comunicació. Aquest tipus d'enllaços tenen una taxa de transmissió pròxima als 64Kbps en ambdós sentits, taxa bastant inferior que en el cas dels enllaços asíncrons. En un mateix canal es permeten fins a tres enllaços simultanis *SCO* i els *slots* temporals que queden lliures poden ser utilitzats per a realitzar enllaços *ACL* de manera que es permet la transmissió simultània de veu i de dades.

Per a garantir la immunitat a interferències, a més a més d'utilitzar la tècnica del salt de freqüència, es garanteix la integritat de les dades gràcies a la utilització del protocol Automatic Retransmission Query (ARQ), que retransmet automàticament els paquets perduts o considerats no vàlids perquè contenen errors. No obstant, aquest protocol no es pot aplicar en sistemes que actuen en temps real —cas de les transmissions de veu— ja que provocaria un efecte no desitjat en recepció. Conseqüentment, s'apliquen diferents esquemes de transmissió basats en la modulació Continuously Variable Slope Delta (CVSD) que permeten que la veu es pugui transmetre fins i tot en entorns amb un nivell de soroll molt elevat ja que els errors en els bits són interpretats com soroll.

Per a la reducció del nombre de retransmissions i limitar l'impacte del soroll aleatori s'utilitzen mecanismes detectors i correctors d'errors. Per a la correcció d'errors en les situacions que el canal presenta nivells de soroll elevats s'utilitza el mecanisme Forward Error Correction (FEC) que pot treballar amb taxes de  $1/3$  —és a dir, per cada bit d'informació se n'envien 3, 1 bit amb la informació útil i dos més de redundants per a poder detectar i corregir errors— i de  $2/3$  depenent del nivell de soroll al que estigui exposat el canal de transmissió. Si el canal presenta nivells baixos de soroll el sistema *FEC* no s'aplica, és a dir, Bluetooth es capaç d'adaptar-se a les condicions del canal en cada moment.

Com es pot veure a la taula 3.3 existeixen diferents tipus de paquets *ACL* i *SCO* en funció del nombre de *slots* ocupats —indicats pel camp *tipus*—, el tipus de *FEC*, utilització o no utilització del Cyclic Redundancy Check (CRC) i de la quantitat d'informació útil que contenen.

### 3.1.3 Topologia de xarxa

Un dels aspectes més interessants de les comunicacions Bluetooth és que es poden crear petites xarxes temporals en les que un nombre limitat de dispositius poden comunicar-se entre si per a intercanviar informació. Existeixen dos tipus de xarxes, les *piconets* i les *scatternets*.

### 3.1. LA TECNOLOGIA BLUETOOTH

Paquet	Tipus	Capçalera (bytes)	Inform. (bytes)	FEC	CRC	Sim. taxa(max) (Kbps)	Asim. taxa(max) TX(Kbps) RX(Kbps)	
ACL	DM1	1	0-17	2/3	sí	108.8	108.8	108.8
ACL	DH1	1	0-27	no	sí	172.8	172.8	172.8
ACL	DM3	2	0-121	2/3	sí	258.1	387.2	54.4
ACL	DH3	2	0-183	no	sí	390.4	585.6	86.4
ACL	DM5	2	0-224	2/3	sí	286.7	477.8	36.3
ACL	DH5	2	0-339	no	sí	433.9	723.2	57.6
ACL	AUX1	1	0-29	no	no	185.6	185.6	185.6
SCO	DM1	na	10	1/3	no	64.0	no	no
SCO	DM1	na	20	2/3	no	64.0	no	no
SCO	DM1	na	30	no	no	64.0	no	no
SCO	DM1	1	10+(0-9)	2/3	sí	64.0+57.6	no	no

Taula 3.3: Característiques tècniques dels paquets Bluetooth

#### Piconet

Si un dispositiu Bluetooth es troba dins del radi de cobertura d'un altre, es pot establir una connexió entre ells compartint el mateix canal. Per a establir un enllaç en principi només calen un parell de dispositius amb les mateixes característiques de hardware. La unió de dos o més unitats Bluetooth que comparteixen un mateix canal forma el que s'anomena una *piconet* (figura 3.3). Com a màxim una *piconet* pot estar formada per 8 dispositius i cadascun d'ells s'identifica per la seva adreça física —una adreça única de 48 bits— identificadora del seu hardware i que el distingeix de qualsevol altre.

Dins de les *piconet* sempre hi ha una única unitat que exerceix de màster —o mestre— i la resta en són esclaus, on el dispositiu que té el paper de màster és el que estableix la *piconet*. El màster s'encarrega de realitzar la sincronització de les comunicacions entre els membres de la *piconet* i de controlar el tràfic de dades del canal, mentre que els esclaus són els que envien i reben paquets al màster i del màster, respectivament.

Per sincronitzar les comunicacions del grup, el màster determina els salts de freqüència i fixa la freqüència de rellotge a la qual s'han de sincronitzar tots els esclaus. Així doncs, quan un esclau vol unir-se a una *piconet* es sincronitza amb el rellotge del mestre i segueix la seqüència de salts que ell ha determinat. Aquests ajustos són temporals ja que quan es tanca o es trenca la connexió, la freqüència de rellotge dels esclaus torna a ser la que tenien originalment.

Per tal de controlar el trànsit del canal, el màster estableix l'ordre de transmissió dins de la *piconet* fixant el torn en el que cada esclau ha de transmetre cada paquet i el comunica a l'esclau durant el *slot* previ al *slot* en el que ha de produir-se la transmissió del determinat paquet. D'aquesta manera s'aconsegueix que no hi hagi col·lisions entre les transmissions dels esclaus.

Els participants poden intercanviar els papers si una unitat esclava vol assumir el paper de màster, però a la *piconet* només hi pot haver un màster a la vegada. Els

dispositius que estan a l'espera de realitzar una connexió es troben en un mode conegut com *standby*, en el qual es pot donar el cas que les unitats esperin una petició de connexió d'algun màster o bé assumeixin el paper de màster i es dediquin a buscar unitats esclaves a les que connectar-se. En el moment de connectar-se es passa al mode *connection*. També existeix el mode *parked*, en el que les unitats es troben sincronitzades amb el màster però sense haver-hi una connexió activa amb ell.

### Scatternet

Una *scatternet* (figura 3.3) és una xarxa formada per dos o més *piconets*. En l'espai de solapament entre elles hi ha unitats que intervenen en la comunicació entre ambdues *piconets*. Les *scatternets* sorgeixen de la limitació de dispositius que tenen les xarxes *piconets* —com a màxim poden estar formades per 8 dispositius— i permeten la interconnexió de les mateixes formant xarxes més extenses i complexes.

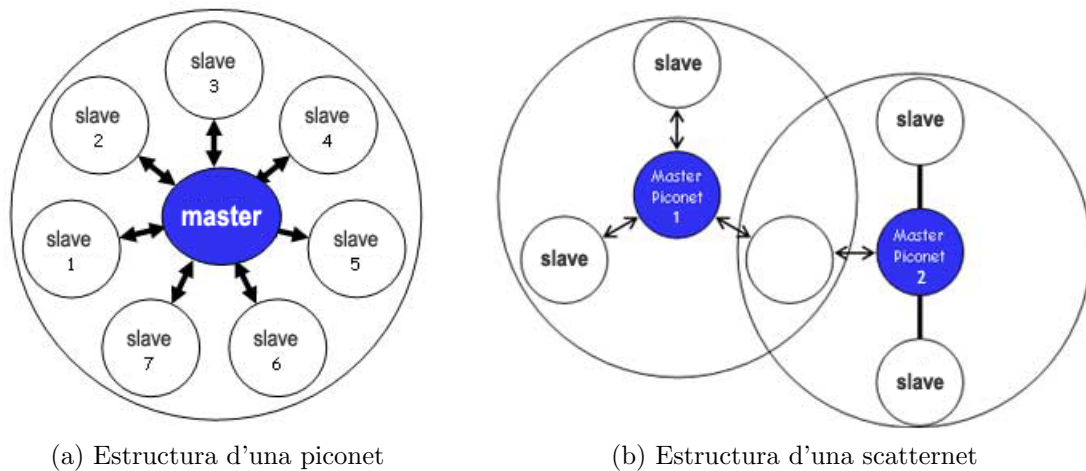


Figura 3.3: Topologia de Bluetooth

Dins d'una *piconet*, els dispositius comparteixen un mateix canal i només poden utilitzar-ne una part de la seva capacitat. Aquesta capacitat disminueix en funció del nombre de dispositius que s'incorporen a la *piconet* —fins a uns 10 Kbps, aproximadament—. En les *scatternets* el rendiment conjunt i individual dels usuaris és major que el que té un usuari quan participa en un mateix canal. A conseqüència de que individualment cada *piconet* té un salt de freqüència diferent fixat pel seu respectiu màster, diferents *piconets* utilitzen diferents canals de salt en un instant determinat. En el cas en el que dos *piconets* diferents comparteixen un mateix canal de salt, els paquets de dades poden ser distingits per la seva capçalera, única per a la *piconet* a la qual pertanyen.

Una unitat participa en una o més *piconets* aplicant multiplexació en el temps —TDD— i per a participar en el canal apropiat utilitza l'adreça del màster associat. Tenint en compte que els rellotges dels màsters de cada *piconet* no estan sincronitzats, una unitat

que participa en dos *piconets* diferents ha de sincronitzar-se al màster de la *piconet* a la qual vulgui participar. Quan una unitat esclava vol canviar de *piconet* ha d'informar al màster que no estarà disponible durant el temps en el que participi amb l'altra *piconet*. Si és el màster el que desitja canviar a una nova *piconet*, ho fa com a esclau de la mateixa i suspen temporalment el tràfic de la *piconet* de la qual procedeix.

#### 3.1.4 Pila de procotols

La pila de protocols de Bluetooth inclou parts hardware, parts software i protocols que es classifiquen en diferents capes tal i com pot apreciar-se en la figura 3.4. Les capes inferiors de la pila —Bluetooth Radio, Banda Base i Link Manager Protocol (LMP)— estan implementades per hardware o firmware, mentre que les superiors ho estan per software.

La capa més baixa és la de radiofreqüència —Bluetooth Radio— i s'encarrega de la transmissió i recepció del senyal radio en la banda *ISM*, utilitzada per l'estàndard Bluetooth. La banda base és la capa que controla la sincronització dels *slots* temporals, el salt de freqüència i la correcció d'errors. El *Link Controller* implementa totes les funcions de la banda base i controla el gestor de l'enllaç o *Link Manager*. El *Link Manager* s'encarrega de la configuració i control de l'enllaç a través del protocol *LMP*, que possibilita la localització d'altres dispositius Bluetooth i la comunicació amb ells, i a més a més s'encarrega dels aspectes de seguretat com són l'autenticació i l'encriptació. Per a realitzar les tasques de configuració i control de connexió entre dispositius, s'encarrega de la negociació de la longitud dels paquets de dades i de decidir els modes de consum de potència o estats de connexió del dispositiu en la *piconet*.

El Host Controller Interface (HCI)[25] és la interfície que separa les capes hardware de les capes software. La seva especificació defineix una interfície de comandes que permet accedir al control de capes inferiors com Banda Base i *LMP* i als estats de hardware i registres de control. D'aquesta manera s'encarrega d'unir el mòdul Bluetooth amb el dispositiu que el controla, per exemple un PC, una PDA o un telèfon.

El Logical Link Control and Adaption Protocol (L2CAP) és el protocol que permet l'accés a les capes inferiors per part dels protocols d'alt nivell i, en general, de les aplicacions que utilitzin comunicacions Bluetooth. S'encarrega de recollir les dades de les capes superiors, adaptar-les i passar-les a les capes inferiors. Les seves tasques més importants són:

- Multiplexació de diverses capes superiors o fins i tot protocols diferents, permetent que comparteixin la mateixa connexió de baix nivell.
- Segmentació i reensamblatge de paquets de dades d'elevat tamany en paquets del tamany adequat per a les capes inferiors.
- Gestió de la qualitat de servei —Quality of Service (QoS)—, assegurant una comunicació adequada entre dispositius Bluetooth.

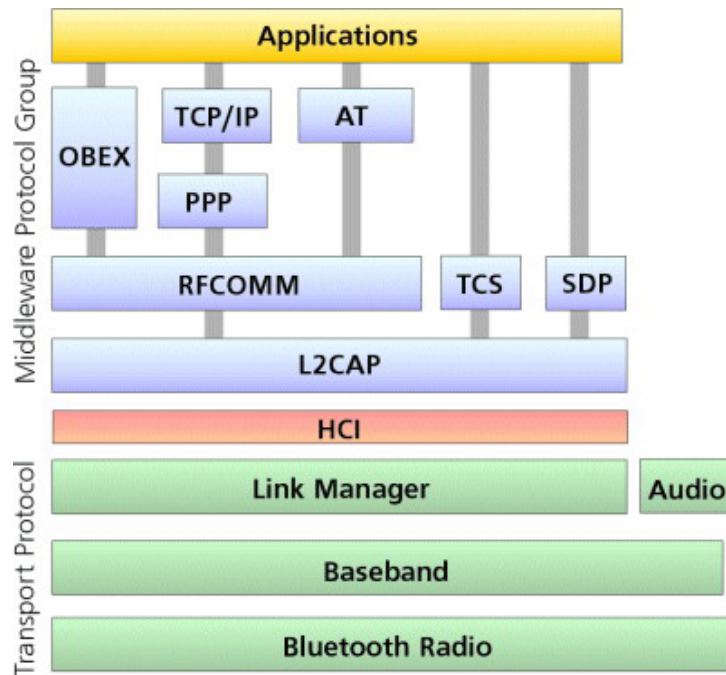


Figura 3.4: Pila de protocols Bluetooth

El SDP és un protocol útil per al descobriment dels serveis disponibles en els dispositius Bluetooth en rang de cobertura. En realitzar una connexió a un dispositiu Bluetooth el que es fa realment és realitzar una connexió a algun dels serveis oferts pel dispositiu remot, per tant és imprescindible esbrinar quins serveis estan disponibles i com realitzar la connexió a cadascun d'ells. Donada la seva importància, la definició de servei i el procés de connexió s'estudien amb més detall en l'apartat 3.1.6.

El protocol RFCOMM proporciona una emulació de ports sèrie a través del protocol *L2CAP*. Es tracta d'un protocol senzill amb suport per a fins a 9 ports sèrie *RS-232*, i que permet fins a 60 connexions simultànies entre dispositius Bluetooth. Per tant, gràcies a *RFCOMM* es pot establir una connexió directa entre dispositius com si es tractés d'una comunicació sèrie cablejada *RS-232*. D'aquest protocol en deriven d'altres de familiars en els quals no s'entra en detall per escapar-se de l'objectiu d'aquest projecte.

Existeixen altres protocols com Telephony Control Protocol (TCS) i OBject EXchange (OBEX), el primer d'ells per a permetre l'enviament de trucades de veu entre dispositius i el segon per a la transferència de fitxers.

### 3.1.5 Perfils

Els perfils, una part primordial en la definició de l'estàndard Bluetooth, tenen com a objectiu principal assegurar la compatibilitat entre les aplicacions Bluetooth i els dispositius dels diferents fabricants. Defineixen la capacitat i paper per a les aplicacions específiques

### 3.1. LA TECNOLOGIA BLUETOOTH

---

de manera que diferents perfils poden abastar diferents capes i protocols. El terme *perfil* va ser proposat per la International Organization for Standardization (ISO), ISO/IEC TR10000.

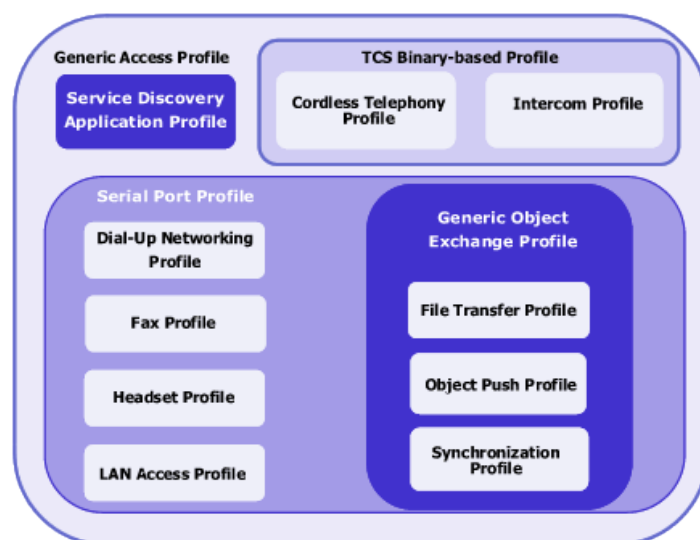


Figura 3.5: Perfils de Bluetooth definits pel SIG

Així, per exemple, per a assegurar que un dispositiu mans lliures Bluetooth d'un determinat fabricant funcioni correctament amb un terminal mòbil d'un altre fabricant és necessari que existeixi un punt de partida o plantilla en comú que assegurï la total compatibilitat i funcionament entre els dos dispositius. Des que es va iniciar l'especificació de l'estàndard, una de les principals preocupacions del SIG fou la de garantir la interoperabilitat total entre dispositius de diferents fabricants sempre i quan aquests compartissin el mateix perfil. És per això que el SIG col·labora contínuament amb empreses i fabricants relacionats amb Bluetooth i les anima per a que puguin establir-se nous perfils que s'ocupin de noves aplicacions Bluetooth.

Els principals perfils existents són els mostrats en la figura 3.5. Tots els dispositius Bluetooth han de suportar un perfil d'accés genèric —Generic Access Profile (GAP)—, que defineix funcions bàsiques com ara el descobriment de dispositius i procediments de connexió, i del qual en deriven tota la resta de perfils. Aquests n'hereden totes les seves característiques i n'afegeixen de pròpies.

#### 3.1.6 Establiment de connexió

Abans que diversos dispositius Bluetooth puguin establir una connexió, és necessari que el dispositiu que pretén iniciar la connexió conegui les característiques dels dispositius Bluetooth disponibles en el seu entorn de cobertura i, sobretot, els serveis que aquests ofereixen per a més tard poder-se connectar a algun d'ells. Un servei no és més que la

funció que ofereix un dispositiu a tot aquell que decideixi connectar-se a ell. Així, per exemple, una impressora amb connectivitat Bluetooth ofereix un servei d'impressió que permet a dispositius externs enviar la informació o arxius a imprimir via Bluetooth.

Per tal de poder establir una comunicació s'han de seguir una sèrie de fases —figura 3.6—, detallades a continuació. En l'explicació es considera que el màster és el dispositiu que inicia la comunicació per a connectar-se a altres unitats esclaves. Les fases són:

**Fase 1: *INQUIRY* (pregunta):** En aquesta fase un dispositiu màster intenta descobrir i identificar quins dispositius hi ha a la seva zona de cobertura. Per a aconseguir-ho, envia un missatge de búsqueda —*INQUIRY*— del que n'espera rebre respostes. Aquest missatge no conté cap tipus d'informació sobre el dispositiu emissor del missatge però pot indicar quina classe de dispositius haurien d'emetre resposta. Per fer-ho, existeix un codi d'accés de pregunta —General Inquiry Access Code (GIAC)— per a preguntar per alguna classe de dispositiu en concret, i una sèrie de codis d'accés de pregunta dedicats —Dedicated Inquiry Access Code (DIAC)— per a especificar encara més el tipus de dispositius buscats. Els codis *GIAC* i *DIAC* es transmeten en diferents freqüències de salt, i una vegada transmesos el màster resta a l'espera de rebre respostes. D'altra banda, el dispositiu que vol ser descobert cada cert temps entra en un estat d'escaneig de preguntes —*INQUIRY SCAN*— amb l'objectiu d'atendre a aquest tipus de missatges. Una vegada atesa la pregunta entra en el mode *INQUIRY RESPONSE* i transmet un missatge de resposta consistent en un paquet Frequency Hop Synchronization (FHS) que conté diferents paràmetres sobre el dispositiu, inclòs un identificador del tipus de dispositiu del que es tracta —PC, terminal mòbil, impressora, etc.—.

**Fase 2: *PAGE* (búsqueda):** És el procediment que segueix al de *INQUIRY* i és on el màster pregunta per l'adreça del dispositiu esclau amb el que vol establir la connexió i es sincronitza amb ell per a poder efectuar la connexió. El dispositiu mestre —que es troba en l'estat *PAGE*— transmet de forma repetitiva i en diferents freqüències de salt el codi d'accés o Device Access Code (DAC) del dispositiu al qual es vol connectar. Degut a que els rellotges del màster i de l'esclau no estan sincronitzats, el mestre no sap exactament quan i en quin canal contestarà l'esclau, amb la qual cosa es queda permanentment a l'escolta fins a rebre la resposta de l'esclau en un canal determinat. L'esclau —en estat *PAGE SCAN*— escaneja el canal i quan identifica el seu propi codi d'accés *DAC* transmet un missatge de resposta —estat *SLAVE RESPONSE*— i es queda en espera. Quan el màster rep la resposta de l'esclau passa al mode *MASTER RESPONSE*, envia un paquet *FHS* amb la seva adreça Bluetooth i informació referent al seu rellotge i torna a l'estat d'espera. L'esclau respon amb un nou missatge que conté la seva adreça Bluetooth i a la vegada es sincronitza amb el màster a partir de la informació continguda en el paquet *FHS* rebut. Aquesta resposta serveix de confirmació per al màster i a partir d'aquest moment pot establir-se una connexió entre els dos dispositius.



**Fase 4: Connexió a un servei:** Una vegada el màster obté la llista de dispositius amb el servei que buscava, pot escollir el dispositiu al que desitja connectar-se per a utilitzar el servei. Quan l'escull, realitza una nova connexió amb la unitat utilitzant informació rebuda en la búsqueda de serveis.

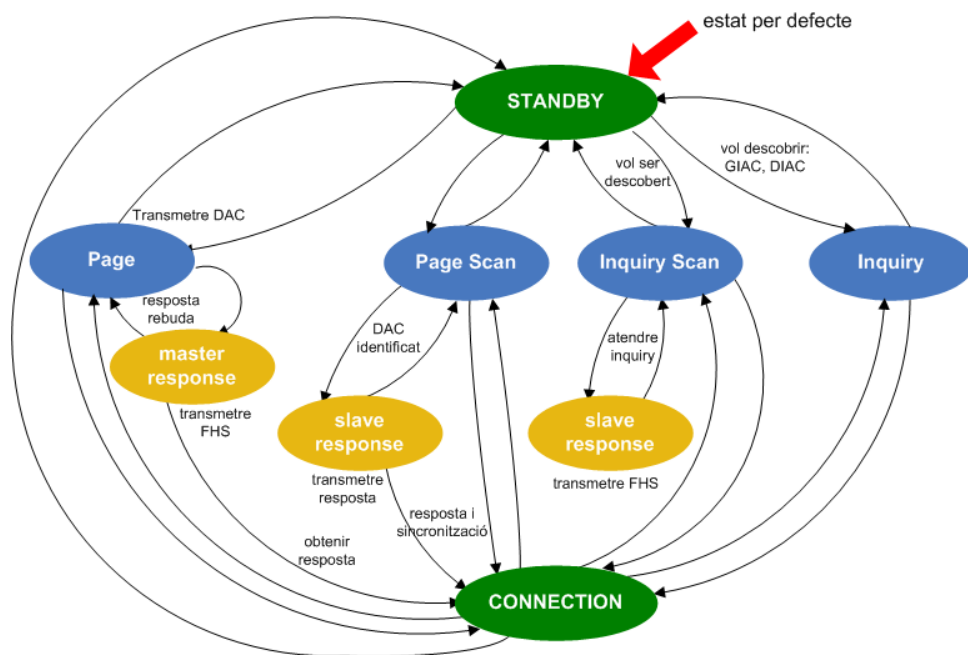


Figura 3.6: Modes de funcionament per a la creació d'una connexió Bluetooth

La tecnologia Bluetooth proporciona una comunicació sense fils entre dispositius, i això comporta un evident risc de seguretat si el tipus d'informació que han d'intercanviar els dispositius és confidencial i aquestes no es protegeixen. És necessari, per tant, establir mecanismes de seguretat que garanteixin la confidencialitat de les dades via ràdio.

Es poden distingir els següents nivells de seguretat:

- a) **Nivell 1 - No segur:** No s'aplica cap tipus de mecanisme de seguretat.
- b) **Nivell 2 - Seguretat imposada a nivell de servei:** El procediment de seguretat s'inicia després que s'hagi establert el canal de comunicació.
- c) **Nivell 3 - Seguretat imposada a nivell d'enllaç:** Es tracta del nivell màxim de seguretat ja que el procediment de seguretat s'inicia abans de l'establiment del canal de comunicació. És el més complex de tots.

A més a més d'aquests nivells de seguretat, n'existeixen d'altres tant per a dispositius com per als serveis que aquests proporcionen. En referència a la seguretat de dispositius, s'han de considerar dispositius de confiança —dispositius que tenen accés sense restricció als serveis— i dispositius de no confiança —tenen un accés limitat als serveis—. D'altra banda, els serveis que ofereixen els dispositius poden ser serveis oberts a qualsevol connexió, serveis que només requereixin autenticació —clau coneguda per ambdós dispositius— o bé serveis que requereixin tant autenticació com autorització —només es permet l'accés als serveis els dispositius de confiança guardats en la base de dades del servidor Bluetooth—.

A continuació es comenta com s'estableix la seguretat a nivell d'enllaç, ja que és el nivell més complex en el que es refereix a seguretat Bluetooth. En aquest nivell hi ha una sèrie de paràmetres que permeten establir una comunicació de confiança entre dos dispositius, i aquests són:

**Adreça del dispositiu Bluetooth:** Identificador de longitud fixa de 48 bits i única per a cada dispositiu Bluetooth.

**Clau d'usuari privat d'autenticació:** Clau de longitud fixa de 128 bits que es facilita durant el procés d'autenticació.

La clau d'autenticació —també anomenada clau d'enllaç— és un número aleatori de 128 bits que és compartit entre dos o més dispositius i utilitzat en qualsevol comunicació entre ells. A més a més, és un dels paràmetres utilitzats per a generar la clau de xifratge. Segons el seu temps d'utilització, existeixen dos tipus de claus d'enllaç: les *temporals* i les *semi-permanents*. Les primeres caduquen en el moment de finalitzar la sessió, és a dir, després que el dispositiu es desconnecti de la *piconet* a la qual estava connectat. Les segones —les *semi-permanents*— són emmagatzemades en memòria no volàtil per a poder ser utilitzades en diverses sessions, i reben aquest nom perquè poden ser modificades.

Les claus d'enllaç es diferencien en diferents tipus en funció del seu mètode de creació i instant d'utilització:

- a) **Clau unitària:** És aquella generada de manera independent en cada dispositiu Bluetooth.

- b) **Clau de combinació:** Formada conjuntament amb la informació d'un dispositiu qualsevol i un altre que participa en la comunicació. Es genera una clau d'aquest tipus per a cada combinació de dispositius.
- c) **Clau temporal o clau mestra:** És la que s'utilitza durant el temps de sessió. Normalment se'n fa ús quan el màster vol comunicar-se amb més d'un esclau utilitzant la mateixa clau d'encryptació.
- d) **Clau d'iniciació:** S'utilitza únicament en el procés d'inicialització, és a dir, quan encara no s'han creat ni intercanviat ni les claus unitàries ni les de combinació. És creada a partir d'un número aleatori, un Personal Identification Number (PIN) i l'adreça del dispositiu Bluetooth. El seu objectiu és protegir la transferència dels paràmetres d'inicialització.

**Clau d'usuari privat de xifratge:** És obtinguda a partir de la *clau d'autenticació*, i és canviada automàticament sempre que s'activa l'encryptació per una comanda del gestor d'enllaç. La separació de la *clau d'autenticació* i la *clau de xifratge* es realitza per a facilitar la utilització d'una *clau de xifratge* més curta i senzilla sense debilitar la seguretat en l'autenticació.

**Número aleatori (RAND):** Ve derivada d'un procés aleatori o pseudo-aleatori que té lloc en el dispositiu Bluetooth, i va canviant amb freqüència.

#### 3.1.8 Alternatives a Bluetooth

Bluetooth no és l'únic sistema de comunicacions inalàmbriques que facilita la interconnexió de dispositius. Existeixen diferents alternatives, cadascuna d'elles amb les seves avantatges i els seus inconvenients segons l'entorn o característiques del servei en el que es vulguin aplicar. Els sistemes inalàmbrics més populars són la connexió per infrarojos o Official Infrared Data Association (IrDA) —una tecnologia inicialment molt extesa en els terminals mòbils però que està entrant en desús—, la més que coneguda Wireless Local Area Network (WLAN) —xarxes locals que segueixen l'estàndard IEEE 802.11— i les novedoses Certified Wireless Universal Serial Bus (W-USB) —la versió inalàmbrica de USB— i *ZigBee* —xarxes personals que segueixen l'estàndard IEEE 802.15.4—.

Tot seguit es passen a analitzar breument aquests sistemes a fi de tenir una idea general de quin és millor en funció de les necessitats. La taula 3.5 mostra un resum comparatiu de les principals característiques de cadascun d'ells.

##### WLAN (IEEE 802.11)

Les xarxes *WLAN*[26] són xarxes inalàmbriques d'àrea local, és a dir, en l'entorn de les *LAN*. Dins de *WLAN* es troben diverses versions que segueixen estàndards diferents, com per exemple els IEEE 802.11b, IEEE 802.11a, IEEE 802.11g i IEEE 802.11n.

Malgrat que ambdós tecnologies treballen en la banda de 2.4GHz —excepte l'estàndard IEEE 802.11a, que opera a la banda de 5GHz— i són omnidireccionals, les xarxes *WLAN* tenen una cobertura superior a les xarxes Bluetooth, denominades xarxes personals in-alàmbriques o WPAN. Així, mentres que les *WLAN* poden tenir radis de cobertura de fins a centenars de metres, amb Bluetooth tan sols s'acostumen a aconseguir rangs de desenes de metres. Aquesta diferència es deu, bàsicament, a que *WLAN* utilitza una major potència de transmissió i per tant té un consum d'energia més elevat. Bluetooth, pel contrari, està orientat a ser utilitzat en dispositius alimentats per bateria on cal tenir un consum energètic reduït. Una altra de les diferències notables entre ambdós tecnologies és que els xips utilitzats per a les comunicacions Bluetooth són molt més econòmics que els de *WLAN*. Respecte la capacitat de transmissió, aquesta és major en *WLAN* que en Bluetooth. Per exemple, en 802.11b, 802.11g i 802.11n es tenen capacitats de 11Mbps, 54Mbps i 108Mbps respectivament mentre que amb Bluetooth aquesta és de 3Mbps, un valor notablement inferior.

Així, doncs, es pot concloure que *WLAN* és útil en els casos en què es necessita tenir un radi de cobertura gran i la potència consumida no sigui tant important. Per contra, si els dispositius que intervenen en la comunicació estan relativament pròxims entre ells i s'alimenten mitjançant bateria és necessari l'ús de menor consum: Bluetooth. En el cas dels telèfons mòbils —dispositius que funcionen amb bateria— tot i que molts models ja incorporen connectivitat *WLAN*, la tecnologia Bluetooth està molt més extesa i és més adequada.

### **IrDA (Infrarrojos)**

Els infrarrojos o *IrDA*[27] és una de les tecnologies que juntament amb Bluetooth incorporen bona part dels terminals mòbils del mercat, tot i que està caient en desús i està sent substituïda per Bluetooth degut a les millores que aporta aquesta.

Amb infrarrojos es tenen radis de cobertura inferiors a 2 metres, un valor inferior que el que ofereix Bluetooth tot i que el consum de *IrDA* és superior. Presenta a més una sèrie d'inconvenients que no té Bluetooth, com per exemple que només poden intervenir dos dispositius en una comunicació. Bluetooth permet comunicacions multipunt, a on un mateix dispositiu pot comunicar-se amb diversos a la vegada, amb la possibilitat de crear petites xarxes temporals. A més a més, en el cas de *IrDA*, és necessari que els dispositius que intervenen en la comunicació tinguin una visió directa, és a dir, que estiguin en línia i entre ells no existeixin obstacles. No obstant, *IrDA* té dos punts forts; un és que pot arribar a tenir capacitats compreses entre els 4Mbps i els 16Mbps gràcies al protocol Very Fast Infrared (VFIR), i l'altre és que les interferències hi tenen una menor presència ja que el seu abast és menor i és més improbable que altres dispositius interfereixin en la comunicació.

En resum, es tracta de dues tecnologies per a l'àmbit de les *PAN* però Bluetooth en general ofereix més avantatges que *IrDA* i d'aquí que a poc a poc l'hagi anat substituint.

### 3.1. LA TECNOLOGIA BLUETOOTH

En el cas del present projecte, per exemple, una solució amb *IrDA* no hagués estat viable per la limitació de visió directa i pel seu curt abast.

#### Wireless USB

*Wireless USB*[28] estableix un protocol de comunicació via ràdio, a partir del ja existent Universal Serial Bus (USB) alàmbic. Opera en un rang freqüencial que va des dels 3.1GHz als 10.6GHz. Permet aconseguir velocitats de transmissió molt superiors a les de Bluetooth, des dels 110Mbps fins als 480Mbps a distàncies que oscil·len entre els 3 i els 10 metres, respectivament. Això ho aconsegueix gràcies a l'ús d'una tecnologia força recent anomenada Ultra Wide Band (UWB). Així doncs, té un radi de cobertura inferior que el de Bluetooth, però el fa ideal per a dispositius multimèdia d'intercanvi d'informació de curt abast i elevades necessitats en quant a capacitat de transmissió com per exemple impressores, discs durs, reproductors d'àudio, etc. El seu consum energètic és superior al de Bluetooth, de manera que el fa menys adequat que Bluetooth per a l'ús en dispositius que s'alimenten per bateria com és el cas dels telèfons mòbils.

<b>sist.</b> <b>paràm.</b>	<b>Bluetooth</b>	<b>WLAN</b>	<b>IrDA</b>	<b>Wireless USB</b>	<b>ZigBee</b>
<b>Aplicació</b>	perifèrics, mòbils	xarxa (internet)	portàtils, mòbils	PC, multimedia	sensors, domòtica
<b>Rang Cobertura</b>	10-100m	100m	1m	3-10m	50-100m
<b>Capacitat</b>	1-3Mbps	11-108Mbps	16Mbps	110-480Mbps	250Kbps
<b>Nodes</b>	7	variable	2	127	65000
<b>Vida Bateria</b>	dies	hores	hores	hores/dies	anys
<b>Usabilitat</b>	mitja/alta	alta	baixa	mitja	baixa
<b>Freqüència</b>	2.4GHz	2.4GHz/5GHz	òptica 850-900nm	3-10GHz	900MHz,2.4GHz
<b>Seguretat</b>	alta	la millor	baixa	mitja	baixa
<b>Cost</b>	baix	alt	baix	mig	baix

Taula 3.4: Diferència entre els diferents sistemes de comunicació inalàmbrica

#### ZigBee

*ZigBee*[29] és el nom de l'especificació d'un conjunt de protocols d'alt nivell de comunicació inalàmbrica per a la seva utilització en radis digitals de baix consum, basada en l'estàndard IEEE 802.15.4 de xarxes inalàmbriques d'àrea personal —*WPAN*—, i està pensat per a ser utilitzat en aplicacions que requereixen comunicacions segures amb baixes taxes de transmissió de dades i maximització de la vida útil de les bateries. És una tecnologia molt similar a la tecnologia Bluetooth però amb algunes diferències, tot i que s'utilitzen en àmbits totalment diferents

*ZigBee* pot constar d'un màxim de 65525 dispositius distribuïts en sub-xarxes de 255 dispositius, mentre que Bluetooth només permet la interconnexió d'un màxim de 8

dispositius en una mateixa *piconet*. Té un menor consum elèctric que el de Bluetooth. En termes exactes, *ZigBee* té un consum de 30mA transmetent i de  $3\mu\text{A}$  en situació de repòs, en front dels 40mA en transmissió i els 0.2mA en repòs que té Bluetooth. Aquest menor consum es deu a que el sistema *ZigBee* roman la major part del temps dormint, mentre que en una connexió Bluetooth aquest fet no es pot donar, i sempre està transmetent i/o rebent dades. Permet aconseguir velocitats de fins a 250Kbps, força inferior que els 3Mbps de Bluetooth.

En quan a la utilització de *ZigBee*, s'ha comentat que està pensat per ser utilitzat en un àmbit totalment diferent al de Bluetooth. Mentre Bluetooth és ideal per a *laptops*, telèfons mòbils o *PDA*s —és a dir, dispositius que funcionen amb bateria però es poden recarregar—, *ZigBee* està pensat per a dispositius que funcionen amb bateria però no són recarregables i que han de transmetre poques dades i molt puntualment, com el cas de xarxes de sensors. De fet, una de les principals aplicacions de *ZigBee* és en l'àmbit de la domòtica. Per tant, tot i la relativa semblança de les tecnologies, Bluetooth segueix sent més apropiat per als telèfons mòbils on els intercanvis d'informació acostumen a ser força constants i llargs.

## 3.2 Descripció del protocol

Tot seguit es descriuen els protocols que s'han dissenyat per al sistema *PPStop*, tant per a la comunicació entre dispositiu electrònic i terminal mòbil —aplicació de control *PPStop*, com per a la comunicació entre PC i terminal mòbil —aplicació de seguiment *PPStop*—.

### 3.2.1 Protocol dispositiu - terminal mòbil

El protocol de comunicació entre el dispositiu electrònic i terminal mòbil s'ha dissenyat tenint en compte les decisions de disseny tècniques i funcionals analitzades en el capítol 2. El comportament general que ha de tenir el protocol es pot representar mitjançant un diagrama de flux —3.7, on amb color vermell es representen les accions inicials i finals, amb groc les accions a dur a terme i en blau els criteris que fan que el protocol prengui una acció determinada o una altra.

#### Posada en funcionament

La primera vegada que es vol posar en funcionament el dispositiu electrònic, aquest s'ha d'activar a través del sensor d'humitat que incorpora. Això genera un event que és captat pel microcontrolador i el fa despertar. Tot seguit s'ha d'engegar el transmissor Bluetooth, i abans que aquest pugui començar la seva activitat primer cal que el microcontrolador el configuri —definir el nivell de seguretat a utilitzar i del nom del dispositiu—. Tot seguit el microcontrolador ha de consultar la llista de terminals *PPStop* que té emmagatzemada

### 3.2. DESCRIPCIÓ DEL PROTOCOL

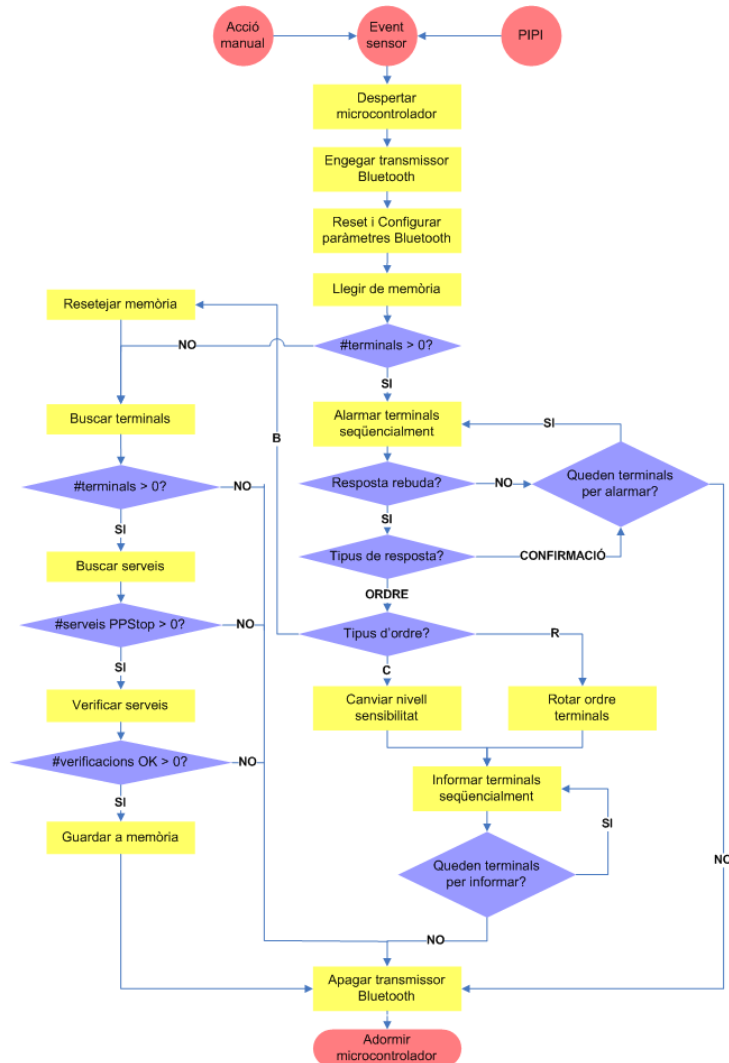


Figura 3.7: Diagrama de flux general del sistema

en memòria. Si aquesta és buida, significarà que el dispositiu no s'ha iniciat mai —o s'ha iniciat alguna vegada però no ha trobat cap terminal *PPStop* dins del seu rang de cobertura—, amb la qual cosa ha de procedir al descobriment de terminals en cobertura mitjançant el protocol *DDP*, prèvia configuració del transmissor Bluetooth —definició del nivell de seguretat a utilitzar i del nom del dispositiu—. En cas que la llista de terminals contingui elements, s'ha de procedir a generar l'avís d'alarma a tots ells.

#### Descobrimient o búsqueda de terminals

Del descobrimient de terminals se n'encarrega el protocol *DDP*. Aquest protocol retorna una llista amb els paràmetres —nom i adreça *MAC*— dels terminals Bluetooth trobats en el seu rang de cobertura, i cal guardar-la en memòria ja que com que només interessen els

terminals que ofereixin el servei *PPStop* a través del perfil *SPP* cal fer una búsqueda de quins són els serveis que tenen actius aquests terminals. Per a això es fa ús del protocol *SDP*, que retorna una llista amb els serveis disponibles en cada terminal de la llista retornada pel protocol *DDP*. Gràcies al *SDP* ja es pot fer una primera discriminació de terminals, i eliminar de la llista aquells que no ofereixen el servei *PPStop*.

No obstant, es pot donar el cas que en aquesta llista hi hagi terminals *PPStop* que pertanyin a un altre dispositiu electrònic i per tant cal fer una segona discriminació. Aquesta es basa en el que s'ha anomenat la *verificació de serveis*, consistent en connectar-se al servei *PPStop* de cada terminal de la llista —paràmetres adreça *MAC* remota i canal d'operació del servei—, enviar una ordre pre-establerta —veure taula 3.5, on DE = Dispositiu Electrònic i TM = Terminal Mòbil— que indiqui a l'aplicació de control que ha de realitzar la verificació del servei i esperar la resposta de verificació de l'aplicació de control. Per a verificar el servei l'aplicació el que fa és comprovar la igualtat entre l'adreça *MAC* origen de la connexió —equivalent al *codi PPStop* proporcionat amb el dispositiu electrònic— amb la que l'usuari ha introduït en l'aplicació —el *codi PPStop*. Tal i com s'ha dissenyat el protocol, s'ha establert que si es dona la igualtat —cas que significa que el terminal *PPStop* correspon al dispositiu electrònic en qüestió— l'aplicació respon a l'ordre de control amb un "OK" i aquest tanca la connexió (es considera aleshores que el terminal ha quedat "associat" al dispositiu electrònic). En cas contrari, no es genera cap resposta i simplement es tanca la connexió. Per tant, cal actualitzar la llista de terminals en memòria i eliminar-ne aquells que no han generat una resposta "OK".

Amb aquest procediment, el dispositiu electrònic ja té en memòria la llista de terminals *PPStop* i per tant, el microcontrolador pot apagar el transmissor Bluetooth i adormir-se fins a la generació d'un nou event per part del sensor d'humitat.

### Generació d'avís d'alarma

La generació d'avís d'alarma és el comportament per defecte del protocol. L'event captat pel sensor d'humitat pot ser degut a dos motius:

- a) El pacient s'està orinant, per tant cal alarmar-lo.
- b) El pacient o un supervisor vol realitzar alguna acció sobre el dispositiu electrònic, com canviar la sensibilitat del sensor d'humitat, rotar l'ordre dels terminals associats al dispositiu o bé iniciar una nova búsqueda de terminals.

Tant en un cas com en l'altre, el dispositiu electrònic sempre actua de la mateixa manera ja que inicialment no sap amb quina finalitat s'ha generat l'event del sensor. Per tant, des del seu punt de vista sempre que rebí l'event del sensor generarà un avís d'alarma als terminals mòbils. El terminal mòbil quan rebí l'alarma s'encarregarà de generar l'alarma sonora o no en funció de les accions que el pacient o supervisor hagi realitzat sobre l'aplicació de control.



### 3.2. DESCRIPCIÓ DEL PROTOCOL

---

Per a generar els avisos d'alarma, el sensor consulta la llista de terminals *PPStop* que té emmagatzemada en memòria i, seguint l'ordre de la llista, va alertant un per un els terminals de manera seqüencial. Per generar una alerta en un terminal, el dispositiu electrònic s'ha de connectar al servei *PPStop* del terminal i enviar una ordre pre-establerta que indiqui a l'aplicació de control que aquella és una connexió d'alarma. En funció del *flag* d'aplicació de l'aplicació de control —*flag* establert segons les accions que el pacient o supervisor hagi realitzat sobre l'aplicació de control—, aquest retorna una o altra resposta —veure taula 3.5—. En cas d'haver servit l'alarma sonora correctament es genera un "OK" de resposta que indica al microcontrolador que ja pot alarmar al següent terminal. En cas de voler realitzar alguna acció sobre el dispositiu electrònic, l'aplicació no serveix l'alarma sonora i genera una resposta en funció de l'acció a realitzar —"*S[SEN]*" per a canviar el nivell de sensibilitat del sensor d'humitat, "*R*" per a rotar l'ordre dels terminals associats al dispositiu i "*B*" per iniciar una nova búsqueda de terminals. Els avisos d'alarma es van generant fins que ja no queden terminals per alarmar, o bé fins que es rep una resposta diferent d'"OK", que atura el procés i fa que el microcontrolador passi a realitzar l'acció ordenada per l'aplicació de control del terminal mòbil. En aquest darrer cas, si l'ordre rebuda és "*S[SEN]*" o "*R*" posteriorment es procedeix a informar a tots els terminals mòbils associats dels nous paràmetres del dispositiu electrònic, és a dir, la possible nova sensibilitat i el possible nou ordre del terminal en qüestió.

El procés de descobriment o búsqueda de terminals és exactament el mateix que el descrit en el punt anterior, amb l'única diferència que abans d'iniciar-lo el microcontrolador fa un reset de la llista de terminals *PPStop* que té emmagatzemada en memòria. El procés de canvi de nivell de sensibilitat consisteix en l'alteració del valor d'alguns pins de sortida del microcontrolador, i s'analitzarà amb més detall en l'apartat 5.3.3 del capítol 5. El procés de rotació de l'ordre dels terminals emmagatzemats en memòria consisteix en rotar circularment els terminals en memòria, de manera que així varia l'ordre en que es generaran les connexions seqüencials.

Tant en un cas com en l'altre, una vegada el microcontrolador ha acabat de fer totes les accions ja pot apagar el transmissor Bluetooth i adormir-se fins a la generació d'un nou event per part del sensor d'humitat.

#### **Transferència de l'estat de la bateria, del nivell de sensibilitat del sensor i de l'ordre del terminal**

Per tal que l'aplicació de control del terminal mòbil pugui mostrar al pacient o supervisor l'estat actual de la bateria del dispositiu electrònic, del nivell de sensibilitat del sensor d'humitat i l'ordre del terminal en la memòria del dispositiu, cal transferir-li aquests paràmetres d'alguna manera. El protocol contempla aquesta necessitat i aprofita per fer aquesta transferència en el moment que envia una ordre a l'aplicació —sigui una ordre de verificació de servei, una ordre d'alarma o una ordre d'actualització d'informació—, afegint aquesta informació tot seguit de l'ordre, i en un format pre-establert —veure taula 3.5—. Això permet que l'aplicació de control rebi una actualització d'aquests paràmetres cada

Dades	Sentit	Descripció
$C[BAT][SEN][O]$	$DE \Rightarrow TM$	Ordre de comprovació de si el servei PPStop es correspon amb el dispositiu electrònic. S'aprofita per enviar l'estat de la bateria [BAT] del dispositiu electrònic, el nivell de sensibilitat [SEN] del sensor d'humitat i l'ordre del terminal en memòria.
$A[BAT][SEN][O]$	$DE \Rightarrow TM$	Ordre d'alarma. S'aprofita per enviar l'estat de la bateria [BAT] del dispositiu electrònic, el nivell de sensibilitat [SEN] del sensor d'humitat i l'ordre del terminal en memòria.
$I[BAT][SEN][O]$	$DE \Rightarrow TM$	Ordre d'actualització d'informació: estat de la bateria [BAT] del dispositiu electrònic, nivell de sensibilitat [SEN] del sensor d'humitat i ordre del terminal en memòria.
$OK$	$DE \Leftarrow TM$	Després d'una ordre de comprovació (C), indica que el servei PPStop es correspon amb el dispositiu electrònic. Després d'una ordre d'alarma (A), indica que l'alarma ha estat servida correctament.
$S[SEN]$	$DE \Leftarrow TM$	Ordre de modificar el nivell de sensibilitat del sensor d'humitat a [SEN].
$B$	$DE \Leftarrow TM$	Ordre d'iniciar una nova búsqueda de terminals.
$R$	$DE \Leftarrow TM$	Ordre de realitzar una rotació en l'ordre de terminals emmagatzemat en memòria.

Taula 3.5: Dades intercanviables dins del protocol DE-TM

vegada que rep una connexió del dispositiu electrònic sense necessitat de realitzar cap connexió addicional.

### Casos particulars del protocol

En la figura 3.8 del final del capítol es pot veure el detall del protocol de comunicació que s'ha descrit anteriorment en el seu cas general. No obstant, es poden donar casos particulars, alguns d'ells ja comentats, i que s'il·lustren en diagrames per separat també al final del capítol com per exemple:

- Durant el descobriment de serveis mitjançant el protocol *SDP*, el terminal mòbil no és un terminal *PPStop*: figura 3.9.
- Durant la verificació del servei, el terminal mòbil no respon —perquè ha apagat la connectivitat Bluetooth o bé perquè ha quedat fora del rang de cobertura— o bé

### 3.2. DESCRIPCIÓ DEL PROTOCOL

Dades	Sentit	Descripció
$T$	$AS \Rightarrow TM$	Ordre de transferir l'historial d'alarmes.
...	$AS \Leftarrow TM$	Transferència de l'historial d'alarmes serialitzat.

Taula 3.6: Dades intercanviables dins del protocol AS-TM

és un terminal *PPStop* amb un servei *PPStop* no correspost: figura 3.10.

- c) Generació d'avís d'alarma a més d'un terminal —2 terminals, per exemple— amb ambdós terminals en cobertura: figura 3.11.
- d) Generació d'avís d'alarma a més d'un terminal —2 terminals, per exemple— amb el segon terminal en cobertura però el primer fora de cobertura, sense connectivitat Bluetooth o amb l'aplicació de control apagada: figura 3.12.

Amb la visió d'aquests casos concrets és fàcil fer-se la idea de com funciona el protocol en qualsevol situació, i en queda totalment descrit el seu disseny. És important notar la presència d'un temps de *TIMEOUT* en totes aquelles accions en què es pot donar el cas que el terminal mòbil no generi cap resposta, sigui pel motiu que sigui. Aquest *TIMEOUT* possibilita que el protocol segueixi el seu curs normal i no es quedi penjat en cap moment.

#### 3.2.2 Protocol aplicació de seguiment - terminal mòbil

El protocol de comunicació entre l'aplicació de seguiment i el terminal mòbil és molt més simple que el vist en l'apartat anterior, i descriu la manera com l'aplicació de seguiment fa la petició de transferència de l'historial d'alarmes a l'aplicació de control del terminal mòbil.

En aquest cas, tal i com mostra la figura 3.13, no es fa ús ni del protocol *DDP* ni del protocol *SDP* ja que l'adreça *MAC* Bluetooth del terminal mòbil i el canal de servei s'introdueixen directament en l'aplicació de seguiment en el moment d'iniciar la importació de l'historial. Això es pot fer gràcies a que l'aplicació de control mostra aquest paràmetre a través de l'apartat de resum de dades. Conseqüentment, doncs, l'única acció que ha de fer l'aplicació de seguiment és establir una connexió Bluetooth amb el terminal i enviar-li l'ordre de transferència d'historial —veure taula 3.6, on  $AS = \text{Aplicació de Seguiment}$  i  $TM = \text{Terminal Mòbil}$ —. L'aplicació de control en rebre aquesta ordre inicia la transferència de l'historial d'alarmes de manera serialitzada, i quan aquesta s'ha acabat de transferir es tanca la connexió. Ja només cal que l'aplicació de seguiment deserialitzi l'historial per a recuperar-lo i poder-lo analitzar per a posteriorment l'aplicació en pugui tractar les dades.

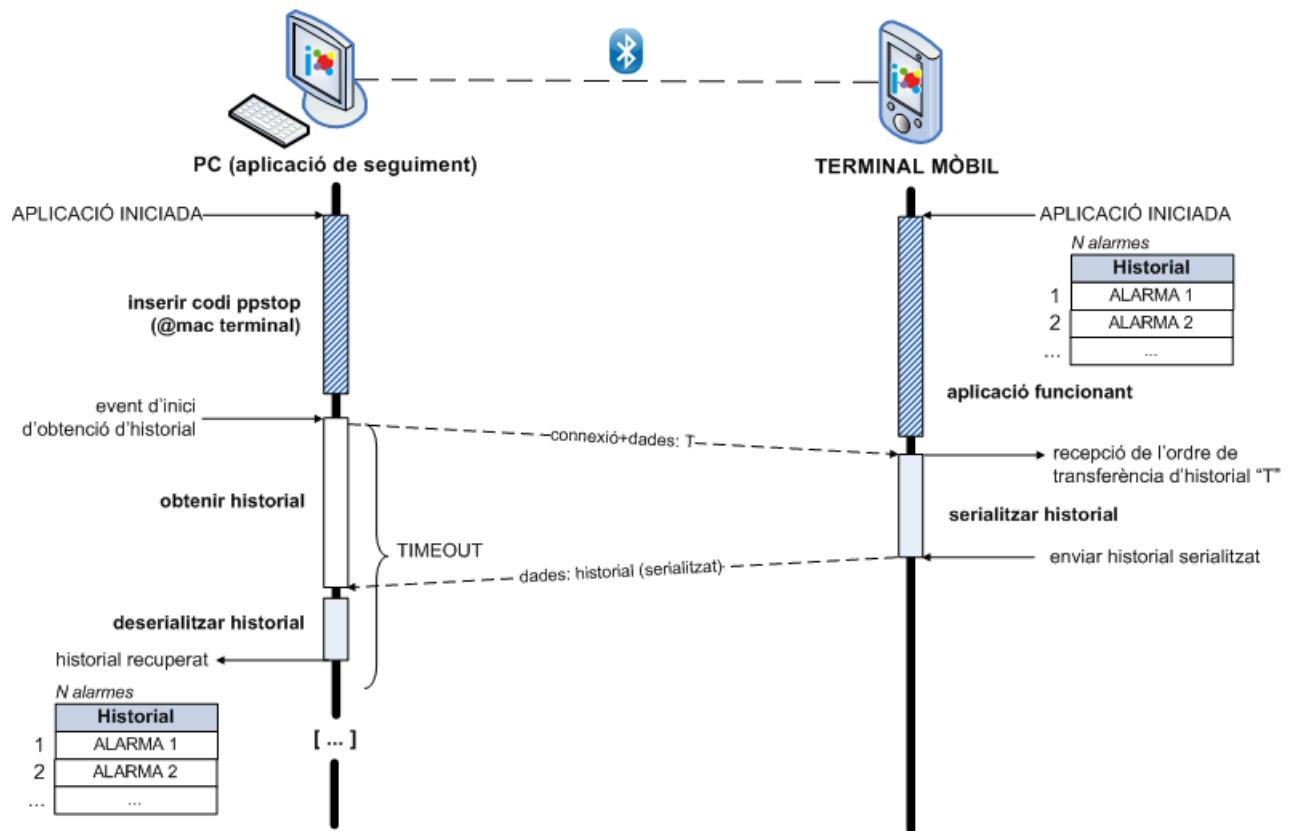


Figura 3.13: Protocol general de comunicació entre aplicació de seguiment i terminal

### 3.2. DESCRIPCIÓ DEL PROTOCOL

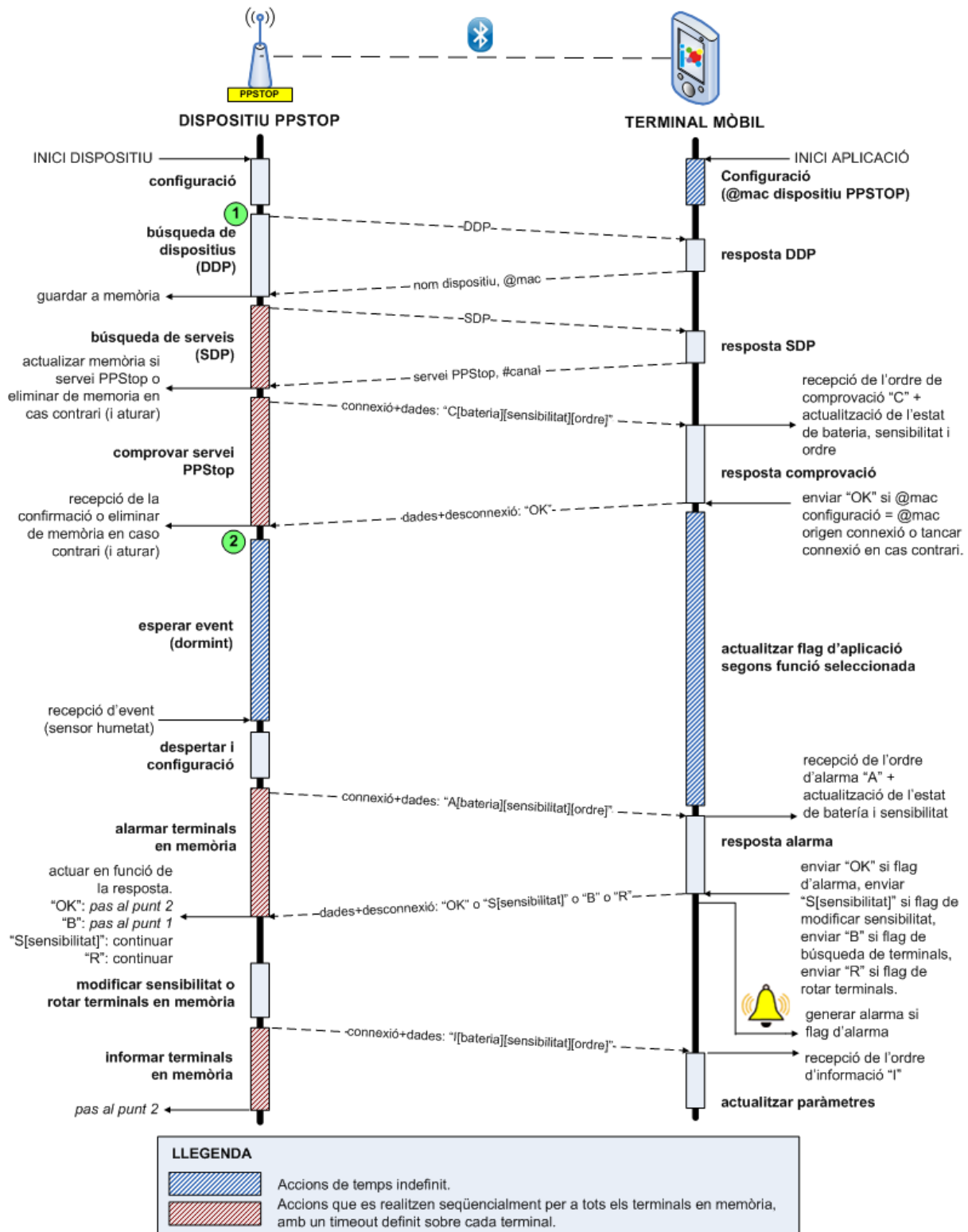


Figura 3.8: Protocol general de comunicació entre dispositiu electrònic i terminal

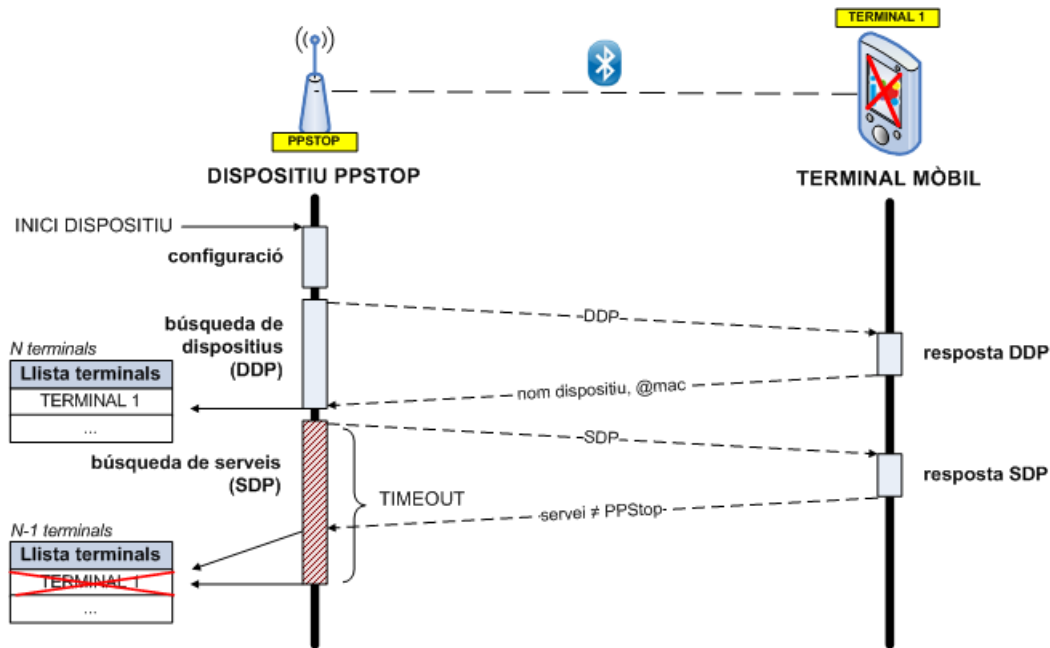


Figura 3.9: Cas concret del protocol: terminal sense servei PPStop

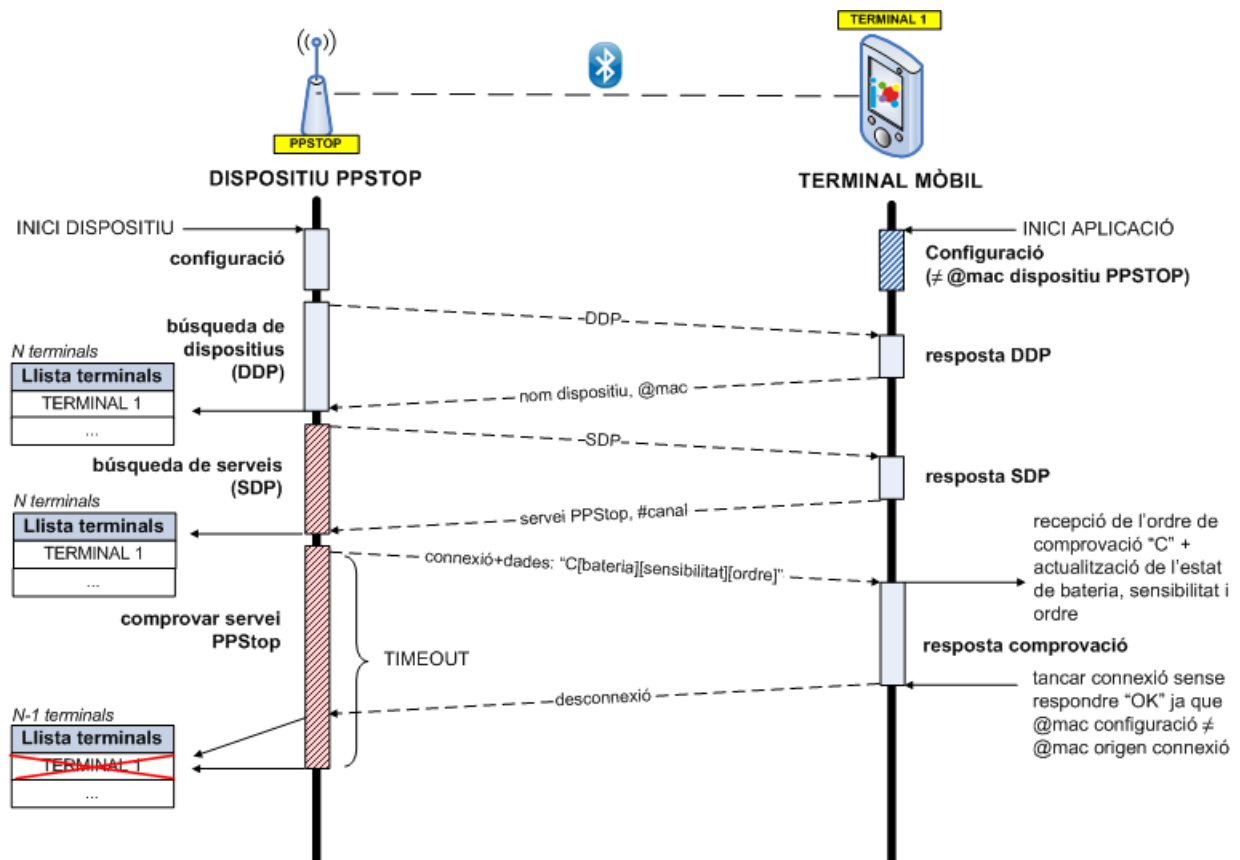


Figura 3.10: Cas concret del protocol: terminal amb servei PPStop no correspost

### 3.2. DESCRIPCIÓ DEL PROTOCOL

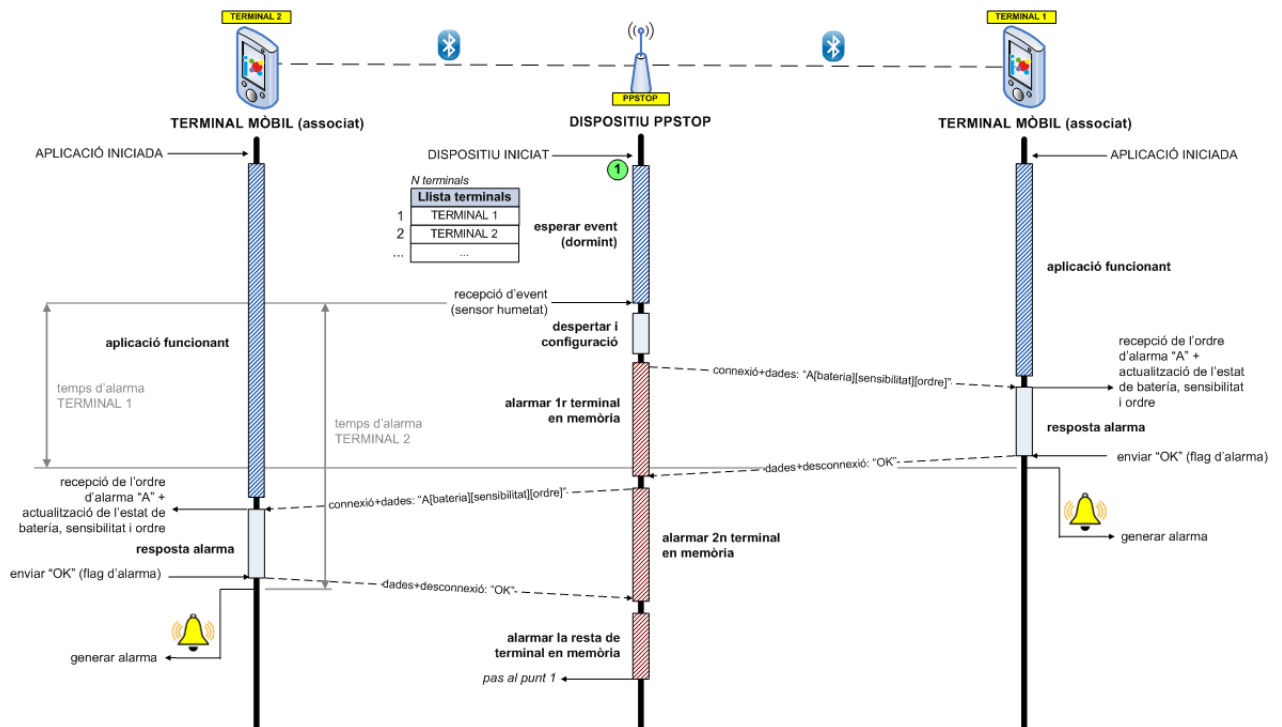


Figura 3.11: Cas concret del protocol: connexió satisfactòria a 2 terminals

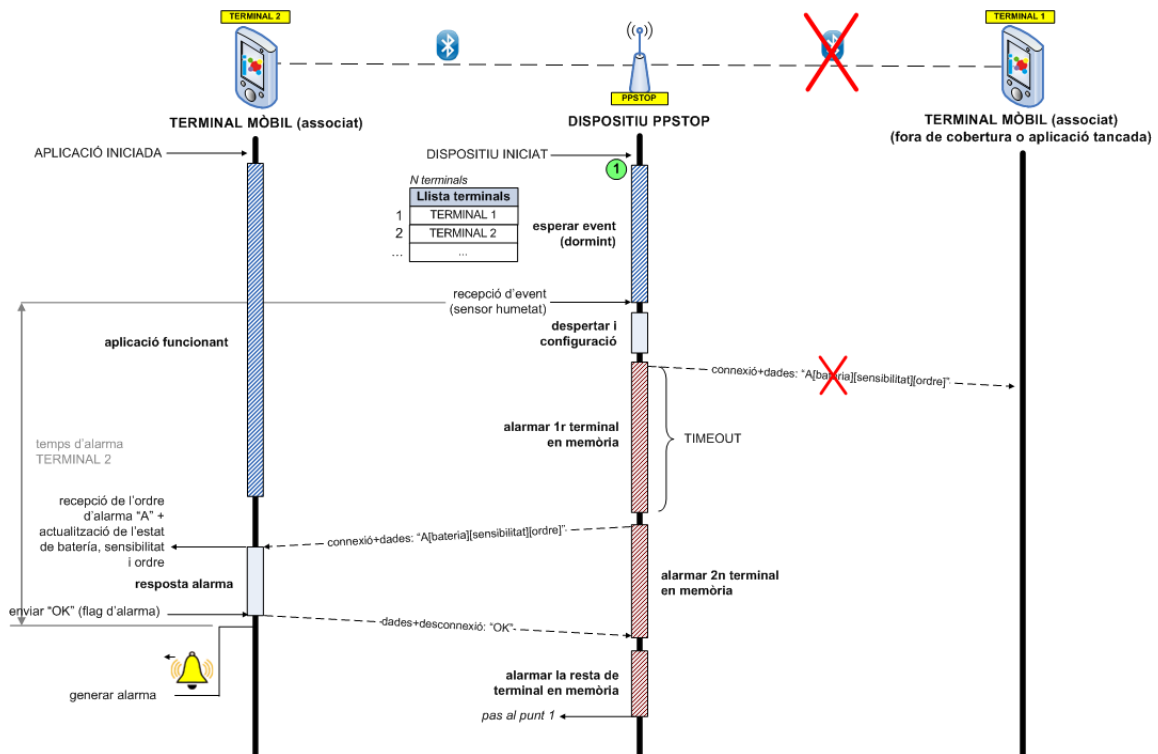


Figura 3.12: Cas concret del protocol: connexió KO per al 1r terminal i OK per al 2n





# Capítol 4

## Aplicació de control

### 4.1 Java 2 Micro Edition (J2ME)

A mitjans dels anys 90, *Sun Microsystems* va desenvolupar el llenguatge de programació *JAVA*. En principi aquest llenguatge estava pensat per al desenvolupament d'aplicacions que controlessin electrodomèstics —frigorífics, rentadores, microones...— però a poc a poc el seu us es va anar estenent en camps com el del desenvolupament d'aplicacions web i el disseny d'aplicacions per a ordinadors. El gran avantatge del llenguatge és que la programació és orientada a objectes i és independent de la plataforma on s'executin les aplicacions, facilitant-ne així la seva portabilitat.

Dins del llenguatge existeixen diferents edicions que es classifiquen en tres grans famílies —figura 4.1— en funció del tipus d'aplicació que es vol desenvolupar i de l'ús que n'ha de tenir:

- **Java 2 Standard Edition (J2SE):** Aquesta edició de *JAVA* conté les eines bàsiques de *JAVA* per a desenvolupar aplicacions que han de ser executades en un ordinador, independentment de la plataforma utilitzada.
- **Java 2 Enterprise Edition (J2EE):** És l'edició de *JAVA* orientada al desenvolupament d'aplicacions en un entorn empresarial de manera que aquestes s'executin de manera remota o distribuïda en una xarxa d'ordinadors empresarial.
- **Java 2 Micro Edition (J2ME):** És l'edició de *JAVA* orientada al desenvolupament d'aplicacions per a dispositius amb capacitat gràfica i computacional reduïdes. Dins d'aquest tipus de dispositius es troben els telèfons mòbils —el cas que ocupa aquest projecte—, PDAs o electrodomèstics intel·ligents.

Cada edició s'adapta a unes necessitats concretes i disposa de llibreries pròpies que permeten satisfer-les, adequant-se al tipus d'equip on s'executen.

En *JAVA*, en general, el codi font és compilat a un codi intermedi —*bytecode*— que ha de ser interpretat per una màquina virtual de *JAVA* —Java Virtual Machine (JVM)—.

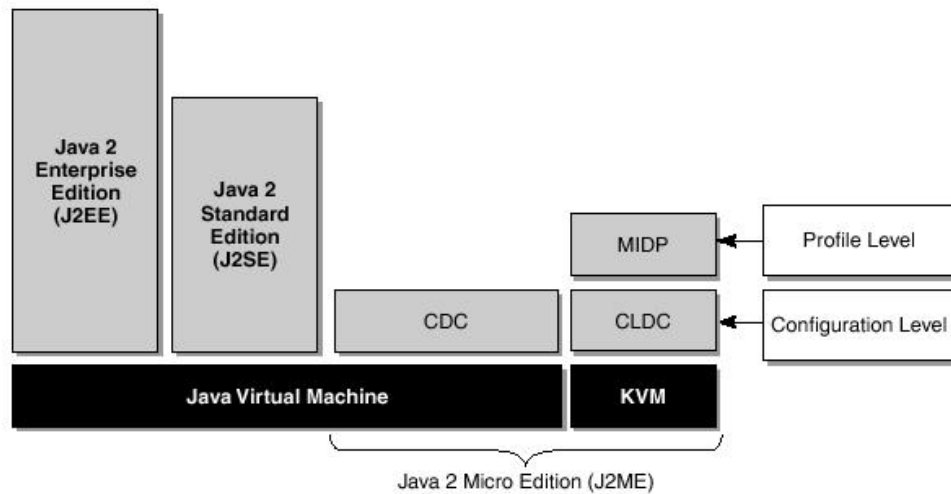


Figura 4.1: Edicions de la família Java

Aquesta màquina virtual ha d'estar instal·lada a la plataforma on s'executa l'aplicació desenvolupada i té un gran avantatge: el codi compilat pot executar-se independentment de la plataforma utilitzada. Una altra de les característiques de totes les edicions és que s'abasten d'*APIs*; llibreries de funcions que satisfan unes determinades necessitats per a la programació d'aplicacions *JAVA*.

En aquest capítol únicament es farà referència als entorns *J2SE*[16] i *J2ME*[6], aprofundint sobretot en aquest últim ja que ha estat el que s'ha utilitzat per al desenvolupament de l'aplicació de control *PPStop* que s'ha d'executar en el terminal mòbil del pacient mentre que *J2SE* només s'ha utilitzat per al desenvolupament del simulador del dispositiu electrònic i en l'aplicació de seguiment del tractament.

#### 4.1.1 Introducció a J2ME

*J2ME* —o *Java 2 Micro Edition*— és l'edició de *JAVA* creada per a la programació de dispositius de recursos restringits com telèfons mòbils i PDAs. *J2ME* es pot veure com una versió simplificada de la edició estàndard de *JAVA* —*J2SE*—, per tant, hereda una petita part de les seves *APIs* i les adapta a la utilització d'aquest tipus de dispositius. D'aquesta manera s'evita utilitzar grans quantitats de memòria com les que són capaces de suportar els ordinadors personals.

Dins de la plataforma *J2ME* es tenen en compte les diferències existents entre els diferents dispositius de recursos limitats i s'estableixen diferents paràmetres per a diferenciar-los —figura 4.2. Així, cada dispositiu disposarà d'un entorn determinat per a l'execució d'aplicacions. Els elements que determinen aquest entorn d'execució en cada dispositiu són:

- 1. Configuració:** Amb les configuracions es defineixen grups de dispositius amb carac-

#### 4.1. JAVA 2 MICRO EDITION (J2ME)

---

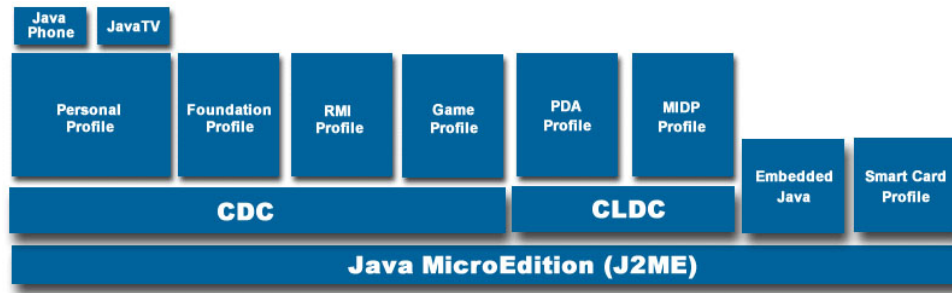


Figura 4.2: Plataforma Java 2 Micro Edition

terístiques comunes pel que fa a capacitat de processador i memòria. Existeixen dos tipus de configuració:

- a) **Connected Device Configuration (CDC)**[30]: Aquesta configuració engloba els dispositius amb més recursos i menys restriccions. Exemples de dispositius *CDC* són els decodificadors de televisió digital, els electrodomèstics i sistemes de navegació per a automòbils.
- b) **Connected Limited Device Configuration (CLDC)**[31]: A aquesta configuració hi pertanyen els dispositius de recursos molt limitats en capacitats de processador i memòria. Exemples de dispositius *CLDC* són els telèfons mòbils, les PDAs, els busca-persones, etc.

**2. Màquina virtual:** La màquina virtual és la base de les aplicacions *JAVA* ja que és l'interpret del llenguatge sobre el qual s'han d'executar les aplicacions. En *J2ME* existeixen dos tipus de màquines virtuals segons el tipus de dispositius en els que s'han d'executar les aplicacions:

- a) **Compart Virtual Machine (CVM)**: És la màquina virtual orientada a dispositius dins de la configuració *CDC*. Suporta les mateixes característiques que la màquina virtual de *J2SE* i està dissenyada per a funcionar sobre dispositius amb processador de 32 bits de gamma alta i amb memòria *RAM* mínima de 2MB.
- b) **Kilobyte Virtual Machine (KVM)**: És la màquina virtual de *JAVA* més petita i està orientada a dispositius dins de la configuració *CLDC*. Tal i com indica el seu nom —*Kilobyte*— permet una càrrega de memòria molt limitada, de l'ordre de desenes de Kb —entre 40 i 80Kb—.

**3. Perfils:** Són biblioteques *JAVA* o *APIs* orientades a implementar funcionalitats de més alt nivell tant per a dispositius que segueixin la configuració *CDC* com la configuració *CLDC*. Per exemple, per a dispositius *CLDC* s'utilitza un perfil anomenat Mobile Information Device Profile (MIDP)[32] que està orientat ser utilitzat en dispositius mòbils i especifica classes que són útils per a l'elaboració d'interfícies d'usuari, emmagatzematge de dades o connexions en xarxa.

**4. Paquets o llibreries opcionals:** Són aquelles llibreries o *APIs* que l'especificació de *J2ME* no estableix com a obligatoris i que proporcionen funcionalitats més concretes. Exemples d'aquest tipus de paquets són les JABWT, la LWUIT o les MMAPI, on el primer afegeix funcionalitats Bluetooth a les aplicacions, el segon permet crear interfícies gràfiques d'usuari més potents i professionals i el darrer permet controlar els aspectes multimèdia dels dispositius.

Aquest capítol centra la seva atenció en la configuració *CLDC* i en el perfil *MIDP* ja que són els que estan directament relacionats amb els dispositius amb màquina virtual *KVM* als quals va dirigida l'aplicació desenvolupada: terminals mòbils. A més a més, es detallen els diferents paquets addicionals a *J2ME* que també han sigut utilitzats per a tal finalitat: *JABWT*, *LWUIT* i *MMAPI*.

#### 4.1.2 Connected Limited Device Configuration (CLDC)

Tal i com s'ha comentat, *CLDC* —figura 4.3— és una configuració orientada a dispositius amb limitacions de capacitat i memòria. Per tant defineix un estàndard per a dispositius amb les següents característiques:

- Memòria total disponible de com a mínim 160KB per a *CLDC* 1.0 i 192KB per a *CLDC* 1.1.
- Processador de 16 o 32 bits.
- Baix consum, ja que està destinat a dispositius portàtils amb bateries de capacitat limitada.
- Connectivitat a algun tipus de xarxa.

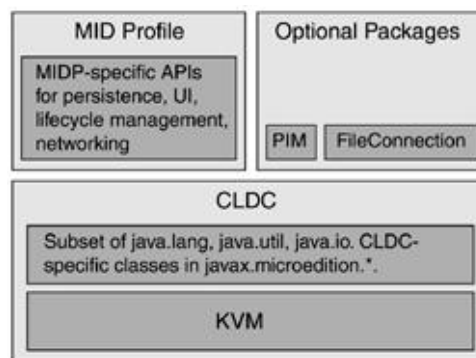


Figura 4.3: Configuració CLDC

Totes aquestes característiques són suportades per la màquina virtual utilitzada en la configuració —la *KVM*— que s'encarrega d'executar les aplicacions i interpretar el codi.

Dins de *CLDC* existeixen dos versions: la 1.0 i 1.1. Com és d'esperar la versió 1.1 ofereix millores respecte la 1.0, cobrint-ne les seves carències i corregint alguns dels seus

errors. Per exemple, entre les seves diferències està el mínim de memòria requerit, sent lleugerament superior per a la versió més nova. A més a més aquest última té suport de coma flotant, entre d'altres característiques.

### Paquets i llibreries de CLDC

Les edicions de *JAVA J2SE* i *J2EE* proporcionen un gran conjunt de paquets amb llibreries molt extenses, però com és lògic per a ser utilitzades es requereixen quantitats de memòria molt grans, de l'ordre de MB. Donades les restriccions de memòria dels dispositius *CLDC*, no és possible utilitzar les mateixes llibreries i s'opera amb un nombre mínim útil per a desenvolupar aplicacions i definir els seus perfils.

La majoria de llibreries *CLDC* són heretades de les de les edicions *J2SE* i *J2EE*. No obstant, degut a les dependències i relacions entre les llibreries originals s'ha optat per tornar a dissenyar-ne algunes i adaptar-les als requisits de *CLDC* de manera que se'n minimitzés el seu nombre i tamany. Així, els paquets *CLDC* es poden dividir en dos grups: els que contenen les classes dels paquets propis de *J2SE* i els que contenen paquets amb classes específiques per a *CDC*, que no provenen de *J2SE*.

Dins del primer grup es distingeix paquets amb classes de sistema, entrada i sortida —E/S— i utilitats (*java.lang*, *java.io* i *java.util*, respectivament). En canvi, dins del segon grup s'hi troba únicament un paquet, *javax.microedition.io*, que proporciona eines per a comunicacions basades en *streams* o en *datagrames* i que conformen el denominat Generic Connection Framework (GCF).

Paquet CLDC	Descripció del paquet
<i>java.lang</i>	Classes i interfícies de la màquina virtual heretats de <i>J2SE</i>
<i>java.io</i>	Classes i paquets per a E/S heretats de <i>J2SE</i>
<i>java.util</i>	Classes i utilitats estàndard heretades de <i>J2SE</i>
<i>javax.microedition.io</i>	Classes per a connexions basades en <i>datagrames</i> o <i>streams</i>

Taula 4.1: Paquets inclosos en la configuració CLDC

### Seguretat en CLDC

Degut a que les aplicacions *J2ME* s'executen en dispositius que emmagatzemen informació personal és convenient que *CLDC* n'asseguri la seva protecció per tal d'evitar que siguin malmesos. Es defineixen tres nivells de seguretat:

- Seguretat a baix nivell.
- Seguretat a nivell d'aplicació.
- Seguretat extrem a extrem.

En la seguretat a baix nivell, el terminal on s'executen les aplicacions es protegeix de possibles danys provocats per codis maliciosos o classes mal dissenyades. Per a tal finalitat, un verificador de fitxers de classes —*Class File Verifier*— s'assegura de que el codi de l'aplicació no conté instruccions il·legals ni referències a parts de memòria no vàlides. És un sistema semblant al que s'utilitza en *J2SE*, però en aquest cas el verificador ocupa un codi binari mínim de 50KB i entre 30 i 100KB de memòria *RAM* i, en el cas de *J2ME*, no és possible utilitzar aquesta quantitat tant gran de recursos. Així, per exemple, en el verificador de fitxers de classes de la *KVM* de *Sun Microsystems* es requereixen tan sols uns 10KB de codi binari i menys de 100bytes de memòria *RAM* en temps d'execució. Per a poder aconseguir aquests valors, els fitxers es pre-verifiquen fora del dispositiu, normalment quan el programador compila el codi de l'aplicació i s'afegeix informació a les classes per a facilitar-ne la posterior verificació.

A nivell d'aplicació, per a garantir la seguretat s'evita que l'aplicació tingui accés en llocs no permesos. Per a aconseguir-ho només es permet l'accés a les llibreries autoritzades per *CLDC* i es prohibeix modificar els mecanismes estàndards de la màquina virtual. Tampoc es permet modificar el grup d'*APIs* que formen el nucli de l'entorn —dins dels paquets *JAVA* i *javax.microedition*—, el perfil i els paquets específics del fabricant del dispositiu.

Per a garantir seguretat extrem a extrem en una comunicació entre dos dispositius, s'utilitzen mètodes com l'encriptació i l'autenticació. Aquest nivell de seguretat queda fora de l'abast de l'especificació *CLDC* i queda sota responsabilitat del programador durant el desenvolupament de l'aplicació.

### 4.1.3 Mobile Information Device Profile (MIDP)

Tal i com s'ha comentat amb anterioritat, *MIDP*[32][33][34] és un perfil construït sobre la configuració *CLDC*. Afegeix un conjunt de llibreries que proporcionen interfícies d'usuari, emmagatzematge persistent, suport a comunicacions de xarxa i jocs. Igual com existeixen dos versions de *CLDC*, també existeixen dos versions de *MIDP* —*MIDP 1.0* i *MIDP 2.0*—. El perfil *MIDP* es troba per sobre de *CLDC* i es per això que l'*API* de les dos versions *MIDP* inclou tots els paquets i llibreries de *CLDC*. Les aplicacions *JAVA* realitzades amb el perfil *MIDP* sobre la configuració *CLDC* reben el nom de *MIDlets*.

#### MIDP 1.0

Es tracta del resultat del treball realitzat pel grup JSR-37 de Java Community Process (JCP), format per experts de diverses companyies. En aquesta versió es defineixen les llibreries que formen l'*API MIDP 1.0* i els mínims requisits hardware que un dispositiu ha de complir. Aquests últims són:

- **Pantalla:** Resolució de 96x54 amb una profunditat de color d'1 bit per pixel.

- **Introducció de dades:** A través d'un teclat o bé pantalla tàtil.
- **Memòria:** 128KB de memòria no volàtil per a les aplicacions *MIDP*, 8KB de memòria no volàtil per a dades persistents de l'aplicació i 32KB de memòria volàtil per a l'entorn d'execució *JAVA*.
- **Connexió a xarxes:** Amplada de banda limitada, bidireccional, sense cables i normalment intermitent.

En referència a les llibreries, cal destacar que *MIDP 1.0* introdueix llibreries i classes addicionals a les que proporciona l'API *CLDC*. Així, per exemple, a més a més de les classes de *javax.microedition.io* especificades a *CLDC* i que conformen el GCF, *MIDP* inclou la interfície *javax.microedition.io.HttpConnection* que possibilita connexions mitjançant el protocol Hypertext Transfer Protocol (HTTP).

### MIDP 2.0

Es tracta del resultat obtingut pel grup de treball JSR-118 de la *JCP*. Aquesta versió 2.0 és compatible amb la de les aplicacions que segueixen el perfil *MIDP 1.0*. Els requisits hardware que s'estableixen són:

- **Pantalla:** Resolució de 96x54 píxels amb 1 bit per píxel de profunditat de color.
- **Introducció de dades:** Igual que en *MIDP 1.0*, a través de teclat o pantalla tàtil.
- **Memòria:** 256KB de memòria no volàtil per a l'aplicació *MIDP*, 8KB de memòria no volàtil per a dades persistents i 128KB de memòria volàtil per a l'entorn d'execució *JAVA*.
- **Connexió a xarxes:** Bidireccional amb accés inalàmbic possiblement intermitent amb amplada de banda limitat.
- **Capacitat de reproducció d'àudio:** És una de les novetats que s'inclouen en la versió 2.0 respecte la versió 1.0.

En la taula 4.2 es poden observar els paquets inclosos en les versions 1.0 i 2.0 de *MIDP*. En aquesta darrera versió s'hi han afegit nous paquets que proporcionen noves funcionalitats respecte la versió 1.0 i també conté altres millores:

- **Especificació d'un gestor d'aplicacions —Application Management Software (AMS)—:** En la versió *MIDP 2.0* es defineixen les especificacions d'un gestor encarregat d'instal·lar, executar o esborrar aplicacions. En la versió 1.0 no hi havia cap especificació d'aquest tipus i el gestor d'aplicacions era implementat per cada fabricant de manera lliure, sense seguir cap mena de pauta comuna.
- **Sistema de descàrrega d'aplicacions Over-the-Air (OTA):** Es permet descarregar i actualitzar aplicacions activament mitjançant connexions a la xarxa. Aquesta especificació, de la mateixa manera que passava amb l'especificació del gestor

d'aplicacions, no havia sigut definida en *MIDP 1.0* i es deixava en mans dels fabricants.

- **Arquitectura *Push*:** Es permet un model de servidor "*push*", que manté la llista d'aplicacions registrades per a rebre informació de la xarxa. Quan arriba informació de la xarxa, el dispositiu decideix quina aplicació utilitzar en funció de les preferències d'usuari.
- **Permisos i firma del codi de les aplicacions:** Es diferencien codis que no són de confiança, on no es pot establir una connexió sense haver rebut permís previ per part de l'usuari, i codis de confiança signats digitalment per entitats certificadores i en els quals l'usuari del dispositiu en pot verificar la signatura. El model de seguretat es millora amb el paquet *javax.microedition.pki*, el qual permet fer ús de certificats que s'utilitzen per a l'autenticació de la informació en connexions segures.
- **Millors en la seguretat de connexions a la xarxa:** *MIDP 2.0* requereix *HTTPS*, *HTTP* sobre Secure Socket Layer (SSL). *SSL* és un protocol que xifra les dades que s'envien per la xarxa proporcionant així una connexió segura.
- **Millors de les GUI:** Possibiliten el desenvolupament d'aplicacions i jocs.

Nom del paquet	Descripció	Disponible a MIDP 1.0	Disponible a MIDP 2.0
<i>javax.microedition.lcdui</i>	Utilitats per a interfície d'usuari	SI	SI
<i>javax.microedition.midlet</i>	Classes per a aplicacions <i>MIDlet</i>	SI	SI
<i>javax.microedition.rms</i>	Persistència de dades	SI	SI
<i>javax.microedition.pki</i>	Seguretat	NO	SI
<i>javax.microedition.media</i>	Utilitats per a àudio	NO	SI
<i>javax.microedition.media.control</i>	Utilitats per al control de l'àudio	NO	SI
<i>javax.microedition.lcdui.game</i>	Utilitats per al desenvol. de jocs	NO	SI

Taula 4.2: Diferències entre els paquets MIDP 1.0 i MIDP 2.0

#### 4.1.4 MIDlets

Tal i com s'ha esmentat anteriorment, els *MIDlets* són aquelles aplicacions *J2ME* que es realitzen utilitzant el perfil *MIDP* sobre la configuració *CLDC*. L'arquitectura de les aplicacions desenvolupades sobre dispositius que incorporen l'arquitectura *MIDP* coexisteix



amb altres tipus d'aplicacions sobre un mateix dispositiu que s'aprofiten de la tecnologia existent. Així, a més a més dels *MIDlets* es distingeixen dos altres tipus d'aplicacions: les aplicacions específiques Original Equipment Manufacturer (OEM) i les aplicacions natives.

Les aplicacions específiques *OEM* són aquelles aplicacions específiques d'un fabricant, realitzades amb classes *JAVA* diferents a la de l'especificació *MIDP*. Per tant, l'inconvenient d'aquestes aplicacions és que normalment no són portables a altres dispositius.

D'altra banda, les aplicacions natives són aquelles que no estan escrites en llenguatge *JAVA* si no que ho estan sobre el software natiu o sistema operatiu del dispositiu. Aquest pot variar en funció del dispositiu. Per exemple, una gran quantitat de terminals mòbils utilitza el sistema operatiu *Symbian OS* i altres, pel contrari, n'utilitzen un de desenvolupat pel propi fabricant.

En el perfil *MIDP* s'estableix quin és l'entorn d'execució d'una aplicació *MIDlet*, quins són els estats pels quals passa i la gestió de recursos per a la seva execució, a més a més de com es realitza l'empaquetatge per a la seva posterior distribució.

### Estats d'un MIDlet

Durant l'execució d'un *MIDlet*, aquest passa per una sèrie d'estats. En el cas de *J2ME* aquests poden controlar-se a través de classes i mètodes continguts en el paquet *javax.micro-edition.midlet*, la classe principal del qual és la classe *MIDlet* i mitjançant la qual es construeixen totes les aplicacions *MIDlet*. Aquesta classe conté els mètodes *startApp()*, *pauseApp()* i *destroyApp()* que permeten el salt entre els diferents estats del *MIDlet* tal i com es mostra en la figura 4.4.

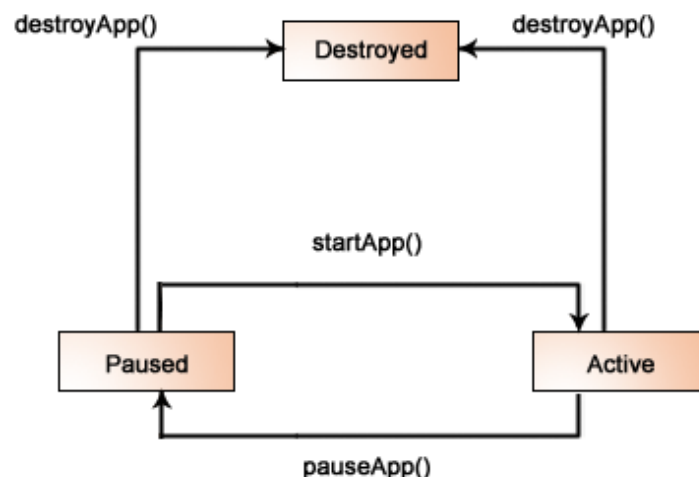


Figura 4.4: Estats d'un MIDlet

Els diferents estats en el que es pot trobar un *MIDlet* són:

- **Detingut:** El MIDlet està inicialitzat i la seva execució ni reserva ni utilitza recursos compartits. Aquest és l'estat en el que es troba l'aplicació quan es crea —fent una crida *"new"*— i encara no s'ha fet la crida al mètode *MIDlet.startApp()*. També pot passar-se a aquest estat després de la crida al mètode *MIDlet.pauseApp()*.
- **Actiu:** En aquest estat el *MIDlet* s'està executant de forma normal i s'hi entra al cridar el mètode *MIDlet.startApp()*, que conté les tasques que el programador desitja que executi l'aplicació.
- **Destruït:** Estat en què el *MIDlet* allibera tots els recursos ocupats —memòria— durant la seva execució i finalitza. Es produeix en fer una crida al mètode *MIDlet.destroyApp()*.

L'aparició d'errors o de l'excepció *MIDletStateChangeException* durant la execució de l'aplicació també pot ser motiu de canvi d'estat automàtic del *MIDlet* i provocar que aquest finalitzi la seva execució.

#### 4.1.5 Empaquetatge i distribució

Els *MIDlets* han d'empaquetar-se d'alguna manera per a facilitar-ne la seva distribució i instal·lació en diferents dispositius. Un o més *MIDlets* s'empaqueten en el que s'anomena una *MIDlet suite* —figura 4.5—. Una *MIDlet suite* consisteix en un arxiu d'extensió Java ARchive (JAR) i una altra Java Application Descriptor (JAD). El primer d'ells conté les aplicacions *MIDlets* en sí, ja que conté les classes compilades que formen cada *MIDlet* en format comprimit. El segon —el d'extensió *.JAR*— no és més que un arxiu de text que actua de descriptor, presentant informació sobre les aplicacions contingudes en l'arxiu *.JAR* —com per exemple, el nom de l'autor, la versió del *MIDlet* que utilitzen, el tamany total de l'arxiu *.JAR* en el que estan contingudes, etc.—.

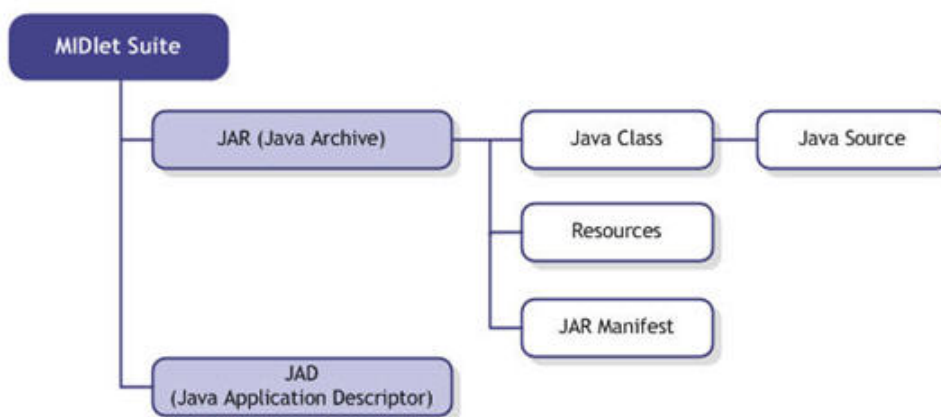


Figura 4.5: Empaquetatge d'un MIDlet

Aquesta informació pot ser obligatòria o opcional segons les especificacions del *MIDP* que s'utilitza en el moment de la instal·lació, càrrega o desinstal·lació dels *MIDlets*. Així

per exemple amb aquesta informació es pot determinar si el dispositiu té memòria suficient per a carregar l'arxiu *.JAR* i executar-lo, comprovar la integritat del paquet o fins i tot saber si es tracta d'una nova versió d'un *MIDlet* instal·lat amb anterioritat. Una altra de les informacions que es pot incloure en l'arxiu *.JAD* és la Uniform Resource Locator (URL) a través de la qual es pot descarregar l'arxiu *JAR* de la *MIDlet suite*.

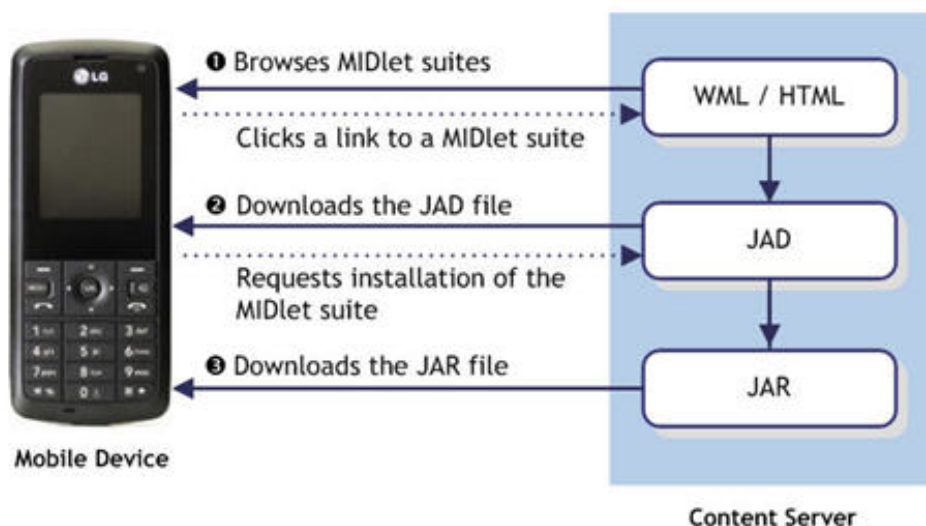


Figura 4.6: Distribució d'un MIDlet

Per a instal·lar la *MIDlet suite* és suficient amb descarregar aquests arxius al dispositiu en qüestió a través d'un cable, una connexió Bluetooth, una connexió WLAN, etc, o bé a través de connexió a Internet mitjançant enllaços que en permetin la seva descàrrega des d'un servidor web —tal i com il·lustra la figura 4.6—. En el cas dels terminals mòbils la connexió a Internet es realitza a través de Global System for Mobile Communications (GSM) —fent ús de navegadors WAP—, GPRS o Universal Mobile Telecommunications System (UMTS). Aquest tema es tracta amb més detall en el capítol 8 en què es descriu aquest procés de descàrrega a través d'una plataforma de descàrrega —en aquest cas a través de SMS—.

### 4.1.6 Record Management System (RMS)

La RMS[35][36] és una *API* de *J2ME* que proporciona un mecanisme a través del qual els *MIDlets* poden guardar dades de forma persistent i recuperar-les posteriorment. Actua com si es tractés d'una base de dades per a terminals mòbils però molt més simple ja que no es permeten fer consultes com sí que succeeix en una base de dades convencional, i a més a més en *RMS* les dades es guarden a nivell de *byte*.

La *RMS* està compresa per múltiples *Record Stores (RS)* tal i com es pot veure en la figura 4.14, i cadascun d'ells es pot entendre com una col·lecció de *Records* —també

anomenats registres—. Un *Record* és un registre individual que permet emmagatzemar una dada individual de qualsevol tipus —*string*, *array*, etc.— en la memòria d'un terminal mòbil. Per a utilitzar *Record Stores* o *Records* cal tenir en compte les capacitats del dispositiu sobre el qual es volen utilitzar, ja que la memòria dels telèfons mòbils acostuma a ser molt limitada.

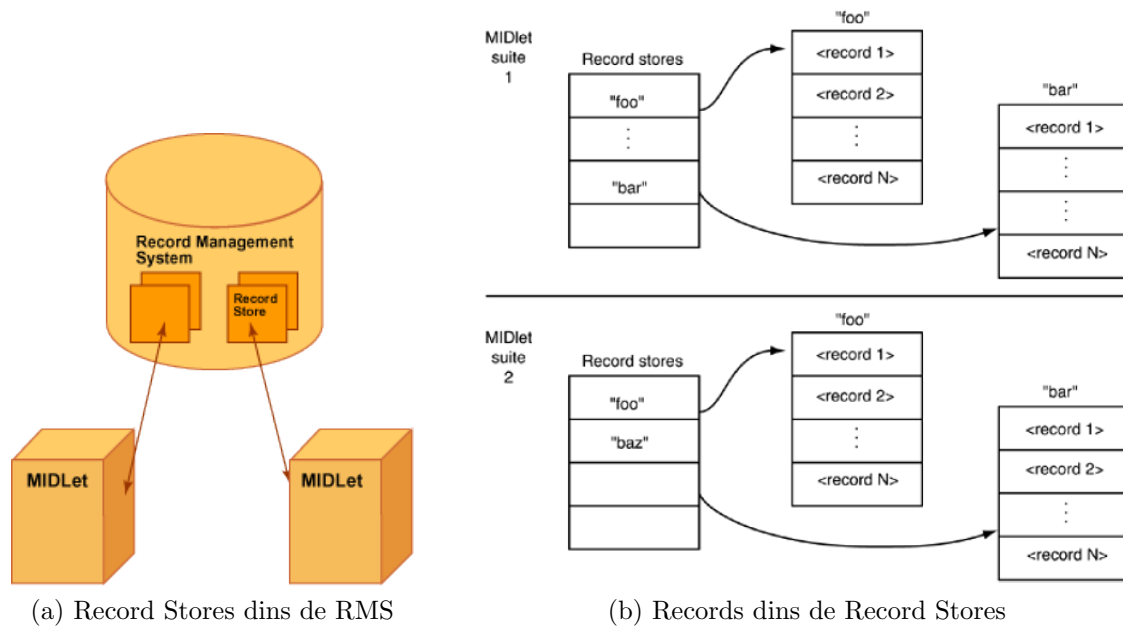


Figura 4.7: Estructura de la Record Management System

A nivell d'API un *Record Store* és representat per una instància de la classe compresa dins de *RMS* `javax.microedition.rms.RecordStore` i totes les classes per a interactuar amb el sistema *RMS* estan definides a `javax.microedition.rms`[37]. Aquestes contenen els mètodes necessaris per descobrir *Record Stores* en el terminal, obrir i tancar un *RS* i inserir, modificar, llegir o esborrar un *Record* d'un determinat *RecordStore*. A continuació es detallen aquests mètodes i quina és la seva funció exacta:

- **RecordStore.listRecordStores():** Obté una llista de tots els noms dels *RS* que emmagatzema l'aplicació a la *RMS*.
- **RecordStore.openRecordStore():** Obre un *RS* i de manera addicional el crea en cas que no existeixi i així se li especifiqui ens els paràmetres.
- **RecordStore.closeRecordStore:** Tanca un *RS* prèviament obert.
- **RecordStore.addRecord():** Afegeix un *Record* en una determinada posició del *RS* i en retorna el seu identificador.
- **RecordStore.setRecord():** Modifica un *Record* amb un cert identificador.
- **RecordStore.deleteRecord:** Esborra un *Record* amb un cert identificador.

Prèviament s'ha comentat que *RMS* emmagatzema les dades a nivell de *byte*. Per tant per a poder emmagatzemar dades d'altres tipus cal fer un mapeig de dades bàsic, altrament conegut com serialització de dades[38][39], i que consisteix en convertir els diferents tipus de dades a emmagatzemar en *arrays* de *bytes*. El procés és el mateix que, per exemple, l'utilitzat en la transmissió de dades Bluetooth a través de l'*stream DataOutputStream*.

Per tal de poder recórrer tots els registres d'un *RS* i recuperar-ne les seves dades, s'ofereixen mètodes addicionals com ara *getNextRecordID()* que salta al següent registre del *RS* i *getRecordSize()* que retorna el tamany d'un registre a partir del seu identificador. Amb aquests mètodes i combinant-los de manera iterativa amb el mètode de lectura d'un registre, es pot fer la recuperació de dades. Cal tenir en compte, però, que les dades recuperades novament tornen a ser *arrays* de *bytes*, i per tant cal fer el procés invers al de serialització per així poder recuperar el tipus primitiu de les dades.

En el desenvolupament de l'aplicació de control *PPStop* s'ha utilitzat *RMS* per guardar la configuració i l'historial d'alarmes de forma persistent en la memòria del telèfon mòbil i així poder recuperar aquestes dades en cada instància d'execució de l'aplicació.

## 4.2 Paquets opcionals per a J2ME

Tal i com s'ha comentat, a banda dels paquets comuns a la edició *J2ME* n'existeixen altres d'opcionals que afegeixen funcionalitats extres. A continuació s'analitzen els paquets extres que s'han utilitzat en el projecte per al desenvolupament de l'aplicació del terminal mòbil.

### 4.2.1 Java APIs for Bluetooth Wireless Technology (JABWT)

Les *JABWT*[40][41][42][7] és un paquet addicional desenvolupat pel grup d'experts del *JSR-82*[43] que afegeix llibreries i classes que permeten establir comunicacions Bluetooth. Està orientat a funcionar sobre dispositius amb configuració *CLDC* de manera que complementin les pròpies de tal configuració.

*JABWT* es divideix en dos subpaquets independents: *javax.bluetooth* i *javax.obex*. La funció i característiques de cadascuna són:

- ***javax.bluetooth***: Proporciona utilitats per a la realització de búsquedes de dispositius i serveis —protocols *DDP* i *SDP*, respectivament— i comunicació mitjançant fluxos de dades o *arrays* de *bytes*.
- ***javax.obex***: Permet la comunicació mitjançant el protocol OBEX, útil generalment per a la transferència de fitxers i també utilitzat sobre altres tecnologies inalàmbriques diferents a Bluetooth.

El sub-paquet *javax.obex* no s'analitzarà ja que no s'ha utilitzat en el desenvolupament del projecte.

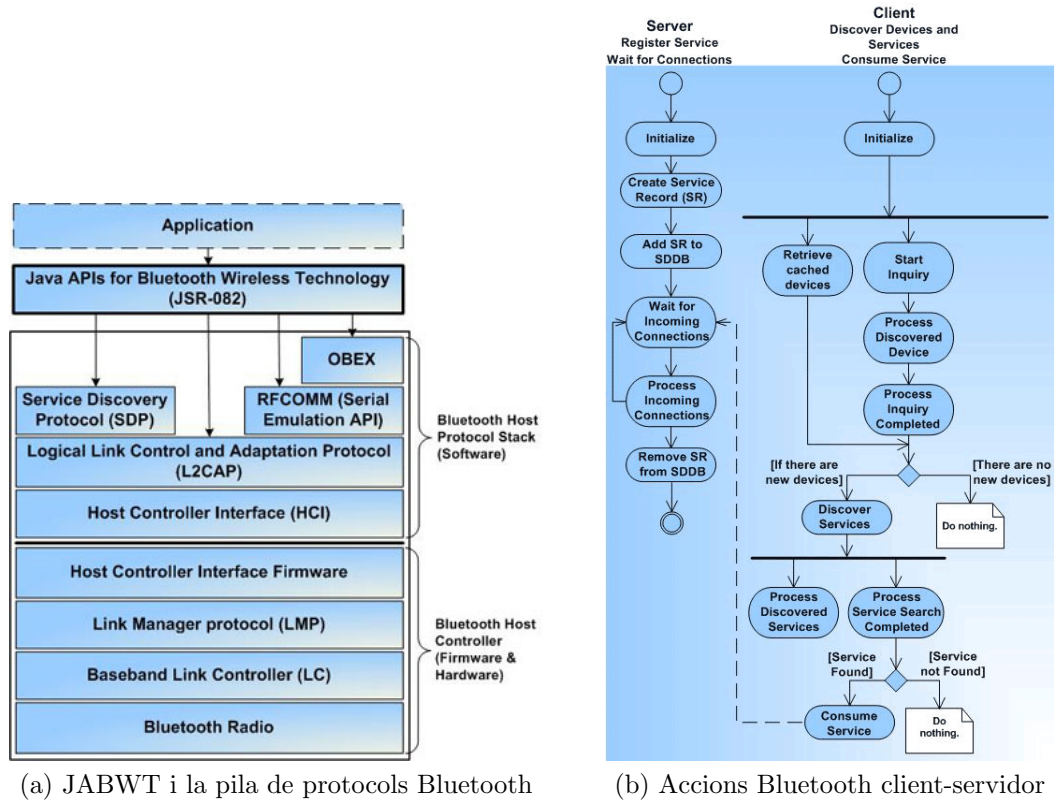


Figura 4.8: Resum de JABWT

## Tipus de comunicacions

El paquet *javax.bluetooth* permet establir la connexió entre dispositius Bluetooth tal i com mostra la figura 4.8, però existeixen dos mètodes principals per a fer-ho: *SPP* i *L2CAP*, que es relacionen amb protocols que ja s'han comentat en el capítol anterior —i resumits també en la figura 4.8—. Per a l'explicació es partirà de la idea que en una comunicació Bluetooth dins del sistema desenvolupat en aquest projecte existeix un dispositiu Bluetooth que actua de servidor —l'aplicació del terminal mòbil— oferint algun servei a altres dispositius —en el cas que ens ocupa, el dispositiu electrònic— que n'esbrina els serveis que ofereix i finalment s'hi connecta.

En les comunicacions *L2CAP* el client i el servidor es comuniquen mitjançant *arrays* de *bytes* de dades. Amb *SPP* en canvi les connexions estan basades en el protocol *RFCOMM*, que ofereix una connexió sèrie sense cables que emula la comunicació sèrie cablejada *RS-232*. En *SPP*, la comunicació es produeix mitjançant *streams* d'entrada i sortida —*InputStream* i *OutputStream*, respectivament— a través dels quals es rep o

s'envia la informació segons el tipus de dades que la conformen —text, enters, bytes, etc.—. *RFCOMM* es troba en una capa superior a *L2CAP* i és per això que *SPP* s'utilitza en la majoria de les aplicacions ja que proporciona eines més completes que s'ajusten més a les necessitats de disseny.

### Creació d'un servidor

Com s'ha comentat, un dispositiu Bluetooth pot convertir-se en un servidor oferint un o més serveis a dispositius remots. Per a oferir un servei, cal crear un *Service Record* i afegir-lo a la Service Discovery Database (SDDB) amb la finalitat que els dispositius en rang de cobertura puguin detectar-lo durant una búsqueda de serveis mitjançant el *SDP*, permetent una posterior connexió per part d'aquests. Per a crear una connexió servidora —crear el registre del servei o *Service Record*— el dispositiu ha de fer ús del mètode *Connection.open()*. Per a poder construir aquest servei és necessari fer-ho a partir d'una *URL* que inclou informació rellevant com ara l'Universal Unique Identifier (UUID) —identificador del servei— i el tipus de connexió que desitja crear —*SPP*, *L2CAP* o *OBEX*—.

L'*UUID* és un enter de 128 bits que identifica a un servei. Ha de ser únic i exclusiu per a un servei de manera que un mateix servidor pugui oferir diferents serveis diferenciats pel seu corresponent *UUID*. Alguns *UUID* es reserven per a aplicacions específiques i la resta queda a disposició dels programadors. A part del *UUID*, a un servei se li pot assignar un atribut que resumeix quin tipus de servei està oferint amb l'objectiu que un dispositiu extern pugui identificar-lo amb més claredat. Un exemple de servei podria ser el servei d'impressió que ofereix una impressora. En aquest cas la impressora podria indicar l'atribut del seu servei si imprimeix a color o en blanc i negre.

Una vegada creat el servei, el servidor entra en espera de rebre peticions de connexió de clients a través del mètode *acceptandOpen()* contingut en les classes *StreamConnectionNotifier* —en cas de comunicacions *SPP*— o el mètode *L2CAPConnectionNotifier* —en cas de comunicacions *L2CAP*—. En aquest moment, el servei —*Service Record*— s'afegeix a la *SDDB* amb l'objectiu de que pugui ser visible pels clients que hagin realitzat una búsqueda de serveis. Amb el mètode *close()* s'elimina el registre del servei de la *SDDB* i aquest deixa de ser visible per a la resta de dispositius.

### Búsqueda de dispositius i serveis

A continuació s'analitzen les classes, interfícies i els seus mètodes que són imprescindibles per a la creació d'una aplicació client que permeti buscar dispositius Bluetooth i els serveis que ofereixen —il·lustrats en la figura 4.9—:

- **LocalDevice:** Aquesta classe representa al mateix dispositiu Bluetooth que intervé en la comunicació —bé sigui client o servidor— i a través d'ella s'iniciarà qualsevol acció.

- **DiscoveryAgent:** Aquesta classe conté els mètodes que possibiliten la búsqueda de dispositius i serveis. S'obté un objecte de la mateixa a partir d'un objecte de la classe *LocalDevice* mitjançant el mètode *getDiscoveryAgent()*. Els mètodes més útils d'aquesta classe són:
  - **startInquiry:** Inicia la búsqueda de dispositius Bluetooth dins del rang de cobertura.
  - **searchServices:** Busca els serveis disponibles —amb uns determinats atributs— en un dispositiu descobert prèviament. Els serveis que ofereix un dispositiu, com hem dit, s'identifiquen amb un *UUID* i un atribut.
  - **cancelInquiry:** Cancel·la la búsqueda de dispositius iniciada amb un *startInquiry*.
  - **cancelServiceSearch:** Cancel·la la búsqueda de serveis en els dispositius descoberts.
- **DiscoveryListener:** Aquesta interfície és capaç de detectar events o successos que s'han anat produint durant la búsqueda de dispositius o dels seus respectius serveis —descobriment d'un dispositiu Bluetooth, descobriment d'un servei, búsqueda de dispositius finalitzada, etc—. Per a cada event hi ha un mètode relacionat que el programador ha d'implementar per a actuar en conseqüència. Els més importants són:
  - **deviceDiscovered:** A aquest mètode s'hi accedeix cada vegada que un dispositiu Bluetooth és descobert.
  - **serviceDiscovered:** Igual que el mètode anterior, però s'hi accedeix cada vegada que es troba un nou servei en un dispositiu Bluetooth.
  - **inquiryCompleted:** S'hi accedeix quan s'ha completat la búsqueda de dispositius —protocol *DDP*—.
  - **servicesSearchCompleted:** Igual però quan finalitza la búsqueda de serveis en un dispositiu en concret —protocol *SDP*—.

En el cas de l'aplicació del terminal mòbil no es faran servir aquests mètodes ja que aquest sempre actua d'esclau —servidor—, i per tant buscar dispositius i serveis no és una tasca que li pertogui, li pertoca al dispositiu electrònic.

### Comunicació entre client i servidor

Per a que un client pugui establir qualsevol tipus de comunicació amb un servidor és necessari l'ús de la classe *javax.microeditions.io.Connector* i del seu mètode *Connector.open()*. Per a realitzar la connexió cal fer-ho a través d'una *URL* que especifiqui el tipus de connexió que volem realitzar —*SPP* o *L2CAP*— i el servidor al qual es vol realitzar la connexió. Així, per a connexions *SPP* la *URL* comença per "*btssp://*" mentre que per a connexions *L2CAP* per "*btl2cap://*".



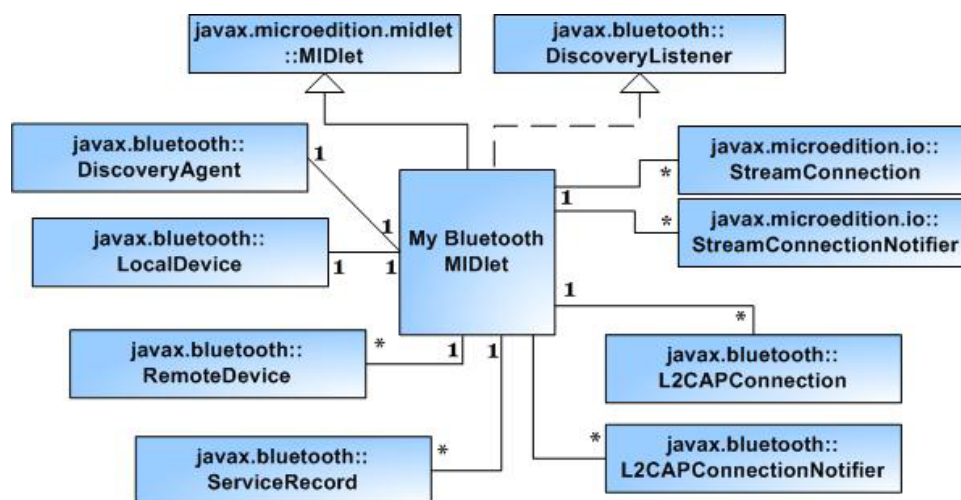


Figura 4.9: Elements d'un MIDlet Bluetooth amb JABWT

Una vegada s'ha establert la connexió, tant el servidor com el client obtenen un objecte —*StreamConnection* o *L2CAPConnection* segons el cas— a través del qual poden iniciar la comunicació. Per al cas de *SPP*, a partir de l'objecte *StreamConnection* es crea un *stream* d'entrada —*DataInputStream*— que permet rebre dades, i un *stream* de sortida —*DataOutputStream*— que permet enviar-ne. Per a tancar la comunicació és necessari tancar-la a tots dos extrems, tancant els *streams* a través del mètode *close()*.

### 4.2.2 Mobile Media API (MMAPI)

El conjunt de llibries de *MMAPI*[9] ha sigut definit pel grup d'experts del *JSR-135* i amplia les eines multimèdia que aporta de per si el perfil *MIDP* per a la creació de *MIDlets*. D'aquesta manera possibilita la reproducció d'àudio i vídeo així com la seva captura i gravació a través del micròfon o de la càmera que incorpora el dispositiu on s'executen les aplicacions. Amb el pas del temps la incorporació, per exemple, d'una càmera de fotos en els terminals mòbils ha sigut més freqüent fins al punt que avui en dia gairebé tots els models disponibles en el mercat n'incorporen una i el seu ús s'ha convertit pràcticament en imprescindible per a l'usuari.

Però entre els diferents models de terminals mòbils presents en el mercat existixen diferències i això també pot observar-se en les seves capacitats multimèdia. Així doncs, pot haver-hi terminals que suportin la reproducció d'arxius d'àudio de diversos formats —*midi*, *wav*, *mp3*, *amr*...— o bé d'altres que només suporten un format en concret. No obstant, cal distingir les capacitats multimèdia que incorpora un mòbil gràcies al seu hardware o software i les capacitats multimèdia disponibles gràcies al *MMAPI* incorporat. D'aquesta manera, un terminal que reproduïx arxius *mp3* gràcies a una aplicació incorporada en el seu sistema operatiu, pot no reproduir-los a través d'un *MIDlet* ja que la seva *MMAPI* no li permet tal operació.

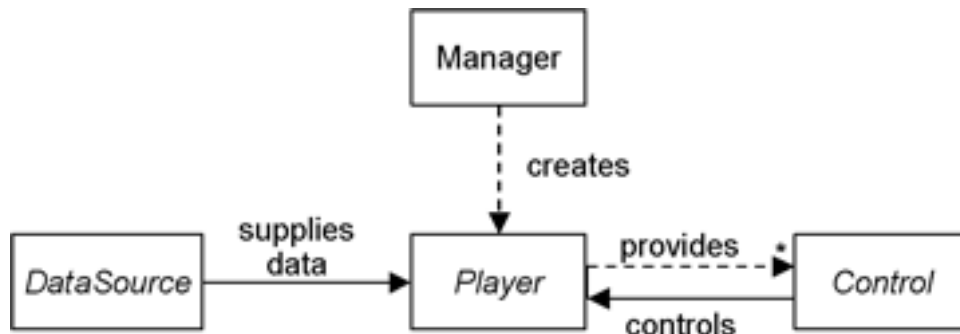


Figura 4.10: Arquitectura i classes de MMAPI

Per tant, les capacitats multimèdia disponibles en un *MIDlet* basat en *J2ME* queden limitades a les característiques del *MMAPI* que el fabricant incorpora en el dispositiu. Conseqüentment també pot passar que dos terminals del mateix fabricant incorporin una càmera però en ser models diferents, amb *MMAPIs* diferents, un permeti la realització de captures de vídeo i l'altre no.

Tot això genera que en el mercat existeixi una gran varietat de terminals mòbils amb diferents capacitats multimèdia. En aquest projecte s'ha desenvolupat una aplicació en la que es reproduïx una alarma sonora quan el terminal mòbil rep una ordre d'alarma del dispositiu electrònic. En el seu desenvolupament, s'ha optat per elegir un format d'àudio que assegura la compatibilitat amb el major nombre de terminals mòbils del mercat i s'adapta a les capacitats dels dispositius mòbils. El format elegit ha estat l'*AMR*[12], un format d'àudio real d'ús molt extès entre mòbils que malgrat treballa amb *bitrates* baixos —compresos entre els 4.75kbps i 12.2kbps— ofereix nivells de qualitat acceptables.

En resum es pot dir que per a que un mòbil sigui compatible amb les opcions multimèdia d'un *MIDlet*, és necessari que incorpori un *MMAPI* adequat. En el cas que ens tracta, per a l'aplicació del terminal mòbil s'ha escollit un *MMAPI* que possibilita la reproducció del format d'àudio *AMR*.

### Classes bàsiques

D'igual manera que el Java Media Framework (JMF) de *J2SE*, el *MMAPI* de *J2ME* presenta unes classes bàsiques sobre les quals es poden realitzar operacions multimèdia. El fet que *J2ME* heredi part de les llibreries de *J2SE* fa que les seves respectives llibreries siguin molt semblants. Tot i això, en el cas de *J2ME* aquestes llibreries s'adapten a les característiques dels dispositius on s'executaran les aplicacions: els dispositius amb capacitat computacional i de memòria limitades.

La classe que serveix com a punt de partida o gestor de les altres de l'*API* és la denominada *Manager* —tal i com mostra la figura 4.10—, sobre la qual no és possible crear-ne una instància i únicament disposa de mètodes estàtics. Amb el mètode *createPlayer()*

s'aconsegueixen crear objectes de la classe *Player*, útils per a la reproducció o la grabació d'arxius d'àudio o vídeo.

Així doncs, en el cas de reproducció, els *Players* reproduïen les dades que els proporciona un objecte *DataSource* associat. Aquest últim llegeix les dades o bé d'un *stream* d'entrada —associat a un arxiu o vector— o bé des d'un localitzador Uniform Resource Identifier (URI) on es trobi l'arxiu, que pot ser per exemple una adreça d'Internet. Per a iniciar la reproducció de qualsevol arxiu abans ha de determinar-se'n la seva codificació o format. En cas que això no fos possible no es pot iniciar la reproducció i es genera una excepció del tipus *MediaException*.

Si l'objecte de tipus *Player* es crea amb èxit existeix la possibilitat de poder controlar el volum de la reproducció, realitzar parades després d'un interval de temps o fins i tot controlar el mode de reproducció d'un vídeo. Aquests controls poden realitzar-se a través del mètode *getControl()* del *Player* creat. El número d'operacions depèn de la implementació del *MMAPI* que contingui el terminal mòbil.

### Cicle de vida de Player

Durant el transcurs d'una reproducció o una gravació d'àudio o de vídeo, i fins la seva finalització, l'objecte *Player* passa per diferents estats —*UNREALIZED*, *REALIZED*, *PREFETCHED*, *STARTED* i *CLOSED*— que formen el seu cicle de vida tal i com es pot veure en la figura 4.11. Per a descriure'ls és prendrà com a exemple el cas de la reproducció d'una alarma, com és el cas d'aquest projecte.

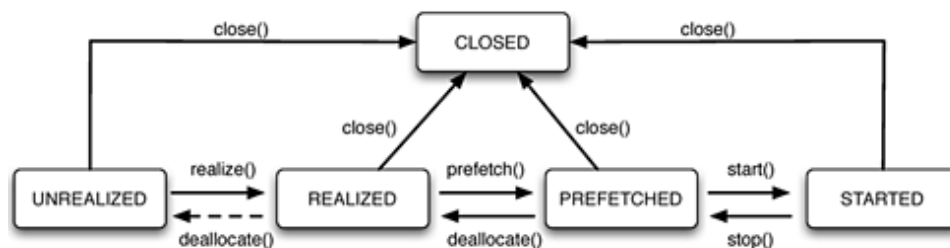


Figura 4.11: Estats d'un objecte de la classe *Player*

Quan es crea un reproductor *Player* aquest inicialment es troba en l'estat *UNREALIZED*. En aquest estat el reproductor no disposa d'informació suficient per a localitzar els recursos o dades necessaris per a començar la reproducció. En el moment en que es crida el mètode *realize()* es recopila tota aquesta informació necessària i es passa a l'estat *REALIZED*.

El següent estat és el de *PREFETCHED*, que es produeix quan el reproductor ha rebut suficient quantitat de dades per a començar amb la reproducció mitjançant el mètode *prefetch()*, de manera que es minimitza el retard entre el moment en el que

s'indica que s'iniciï la reproducció a través de la crida del mètode *start()*, i l'instant en el qual realment comença. En el moment en el que s'inicia la reproducció es passa al següent estat: *STARTED*. Per a tancar el reproductor i alliberar els recursos que aquest ocupa, es realitza una crida a *close()* i, en conseqüència, es passa a l'estat *CLOSED*.

Conèixer els estats en el que es troba el reproductor és de molta utilitat si es volen realitzar determinades accions per a estats concrets. Per exemple, si es desitja realitzar la reproducció continua d'un mateix arxiu és necessari saber quan s'arriba al final de la seva reproducció per a que en aquell mateix punt la reiniciem. Per a tal finalitat existeix la interfície *PlayListener*, amb la qual es pot implementar un mecanisme de detecció d'events generats pels reproductors que permet conèixer el seu estat i actuar en conseqüència. Dins del mètode *playerUpdate()* es pot identificar l'event en qüestió i realitzar l'acció que més convingui. Exemples d'events poden ser la finalització d'una reproducció d'un arxiu —*END OF MEDIA*— o la parada de la reproducció —*STOPPED*—.

### 4.2.3 Lightweight UI Toolkit (LWUIT)

Un dels punts dèbils que tenen les especificacions de *J2ME* és la pobresa que ofereix la seva *API* de construcció d'interfícies gràfiques, que té moltes limitacions a l'hora de crear interfícies professionals i amb una mica d'estètica. Una possible opció és programar directament sobre el *canvas* del *MIDlet* o utilitzar *APIs* propietàries de cada fabricant de terminals o altres llibreries de pagament. Donada la dificultat de programar sobre el *canvas* i per a cobrir aquesta deficiència de *J2ME* es va crear *LWUIT*[10], una llibreria addicional d'ús gratuït per a ajudar als desenvolupadors d'aplicacions per a terminals mòbils en la creació d'interfícies gràfiques de qualitat.

Aquesta llibreria va ser creada per *Chen Fishbein* en el *Centre de Desenvolupament de Sun Microsystems a Israel (SIDC)* com a projecte intern, però no va ser fins a l'any 2008 que va començar a adquirir una gran acceptació dins de la comunitat *J2ME*. L'*API* està profundament inspirada amb *SWING*[11] —l'eina oferta per *Sun Microsystems* per a la creació d'entorns gràfics en *JAVA*— i es basa en una arquitectura formada per components i contenidors, que poden ser combinats per a l'obtenció de layouts totalment adaptats a les necessitats del programador.

*LWUIT* requereix *CLDC 1.1* o superior i *MIDP 2.0* o superior, i permet la creació d'interfícies amb suport per a temes gràfics, transicions, animacions i altres. És una *API* molt fàcil de comprendre i està molt ben documentada[44]. Amplia la majoria dels controls gràfics estàndard del paquet *javax.microedition.lcdui* amb la incorporació de controls especials com ara els *ComboBox*, els *ListBox*, els *TabbedPane*, els *Dialog* o els *Calendar* —veure figura 4.12— a més a més d'oferir una bona gama de layouts —*GridLayout*, *BoxLayout*, *BorderLayout*, etc.— que donen molta versatilitat al programador.

A més a més, incorpora suport per a pantalles tàctils, suport per a l'aplicació de temes gràfics en temps d'execució i suport per a màscares d'entrada per als camps d'entrada de text, entre moltes d'altres sorprenents característiques, de manera que l'interfície gràfica

### 4.3. DESENVOLUPAMENT DE L'APLICACIÓ

---

es pot personalitzar totalment donant un aspecte gràfic únic a l'aplicació a canvi d'un petit increment en l'ús dels recursos del terminal mòbil.

En el cas de l'aplicació de control *PPStop* s'ha fet ús d'aquesta *API* donada la possibilitat que en un futur l'aplicació pugui ser un producte al mercat i, per tant, era important que l'aplicació cuidés la seva imatge de cara l'usuari final.

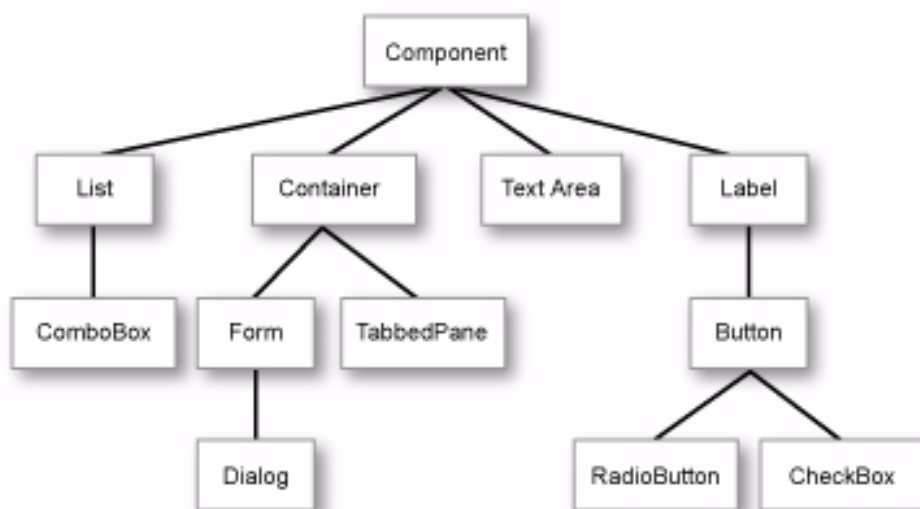


Figura 4.12: Jerarquia de components de LWUIT simplificada

## 4.3 Desenvolupament de l'aplicació

En aquest apartat es descriu l'estructura que s'ha seguit en el desenvolupament de l'aplicació del terminal mòbil —aplicació de control— amb el llenguatge *J2ME* les característiques del qual s'han analitzat anteriorment.

Durant el seu desenvolupament, s'han consultat de manera regular les referències [45],[46],[47],[48],[49],[50],[8],[51] i [52] en referència a la programació general en *J2ME* i la relacionada amb Bluetooth; [53],[54],[10] i [44] en la programació de l'interfície gràfica amb *LWUIT*; [55] per a la programació de la reproducció d'àudio; i d'altres ja esmentades durant aquest capítol per a la utilització de les *APIs* de *J2ME* en general així com del sistema d'emmagatzematge de dades *RMS*. A més a més han sigut especialment útils les consultes esporàdiques en els fòrums per a programadors d'aplicacions per a dispositius mòbils dels fabricants *NOKIA*[56] i *SONY ERICSSON*[57].

### 4.3.1 Entorn de desenvolupament

Per a la programació amb el llenguatge *J2ME* s'ha fet servir l'entorn de programació *NetBeans*[58] —versió 6.7— juntament amb el paquet Java 2 Standard Development Kit (J2SDK) —versió 1.4.2.9— que també ofereix gratuïtament *Sun Microsystems* a través de la seva pàgina web. Existeixen altres tipus d'entorns de programació, però el principal avantatge que té *Netbeans* és que està disponible tant per a la plataforma *Windows* com *Linux* i proporciona eines i mòduls excel·lents que faciliten molt la vida del programador, i més concretament en el desenvolupament d'aplicacions *JAVA*.

El paquet J2SDK instal·la una màquina virtual *JAVA* i les llibreries de *J2ME* necessàries per al desenvolupament d'aplicacions amb aquest llenguatge. A més a més, també s'han afegit les llibreries *JABWT*, *MMAPI* i *LWUIT* analitzades en els apartats 4.2.1, 4.2.2 i 4.2.3 respectivament i que ha calgut utilitzar donades les necessitats de l'aplicació.

### 4.3.2 Patró Model-Vista-Controlador (MVC)

El patró MVC[59] és una arquitectura de software molt utilitzada en l'enginyeria del software. Defineix un estil de programació que separa l'aplicació en tres elements bàsics que interactuen entre si: el model, la vista, i el controlador. El fet de fer aquesta separació ajuda a obtenir un millor rendiment de funcionament de les aplicacions, en simplifica molt la programació i ajuda a resoldre alguns dels típics problemes amb els que es troba un programador en la fase de desenvolupament d'una aplicació, com per exemple la gestió de varies interfícies gràfiques o diversos models de dades en una mateixa aplicació, o l'actualització de la interfície gràfica quan les dades de l'aplicació canvien. Tot seguit es defineix el paper que juga cadascun dels tres elements existents dins del patró:

**Model:** Representa les dades o la lògica de negoci de l'aplicació. Per exemple, en una aplicació de comptabilitat, el model seria la part de l'aplicació encarregada d'anar a buscar les dades als components o capes més baixes, manipular les dades, fer els càlculs que fossin necessaris i notificar a les vistes enregistrades al model que un tipus determinat de dada ha sofert canvis.

**Vista:** És la interfície —gràfica o no— encarregada de mostrar les dades i les comandes relacionades a l'usuari. Donat un únic model, algunes aplicacions ofereixen múltiples vistes sobre unes mateixes dades. En resum, s'encarrega de presentar el model en un format adequat per a que l'usuari pugui interactuar amb elles i generalment ho fa mitjançant una interfície d'usuari.

**Controlador:** És l'encarregat de respondre als events, habitualment accions de l'usuari, i segons l'event desencadenat invoca peticions al model i probablement a la vista. Pot estar centralitzat o repartit en diverses classes on cadascuna d'elles es responabilitza d'una part determinada de la lògica de l'aplicació.

### 4.3. DESENVOLUPAMENT DE L'APLICACIÓ

Aquesta separació implica, en el cas de *JAVA*, separar i organitzar les classes segons la seva funció [60][61]. Hi haurà classes que s'encarregaran de generar la vista, n'hi haurà que controlaran els events de l'aplicació, n'hi haurà que tractaran i manipularan dades i n'hi haurà també que facin funcions concretes que no es puguin classificar en cap d'aquests tres tipus.

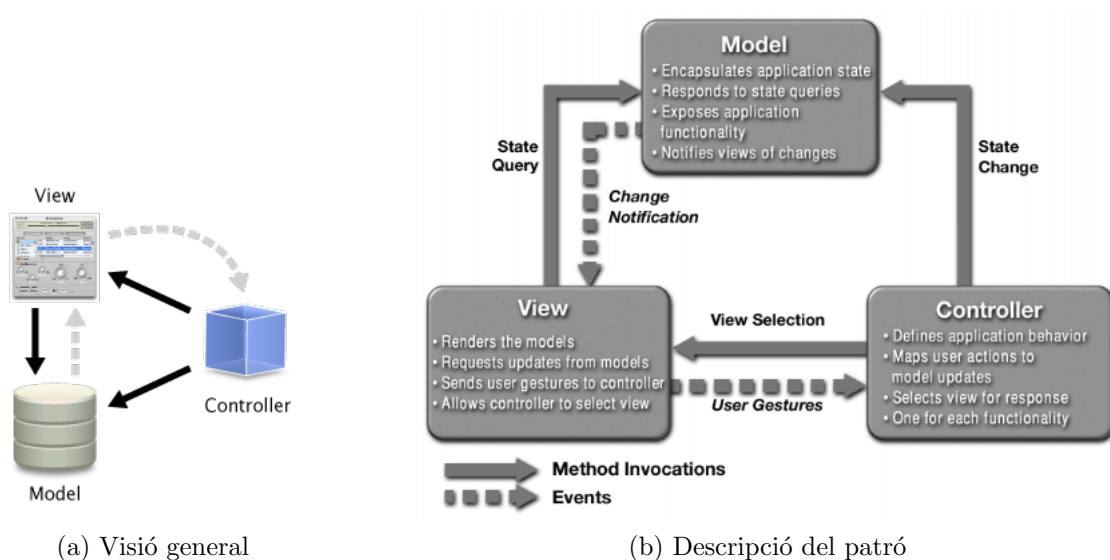


Figura 4.13: Patró Model-Vista-Controlador

Un exemple simple de funcionament del patró es pot veure en el diagrama de la figura 4.13. La vista conté la interfície gràfica d'usuari —*GUI*—, que és la part de l'aplicació que l'usuari pot veure. Quan l'usuari altera alguna dada de la pantalla i fa alguna acció, aquesta genera un event en el controlador, qui el capta i determina el tipus d'acció que ha de portar a terme —lògica de l'aplicació—. Aquesta pot desencadenar una crida a alguna funció del model que n'actualitzi la dada modificada per l'usuari, o directament un canvi de la vista que té registrada. En el cas que el controlador cridi una funció que modifica alguna dada del model, posteriorment el model ho notifica a la vista per tal que aquesta s'actualitzi i mostri la dada modificada a l'usuari.

En el cas de l'aplicació *PPStop* del terminal mòbil s'ha seguit aquest patró de programació, tal i com es mostra en el següent apartat.

#### 4.3.3 Estructura i funcionament

La principal funció de l'aplicació del terminal mòbil —aplicació de control *PPStop*— és la de rebre els avisos d'alarma generats pel dispositiu electrònic a través d'una connexió Bluetooth i generar una alarma sonora en el terminal del pacient per tal de que es desperti quan aquest s'orina. Per tant, actua principalment com a dispositiu d'alarma. No obstant, i tal com s'ha descrit en els capítols 2 i 3, també ha de ser capaç d'enregistrar les alarmes

generades en un historial d'alarmes i de poder modificar certs paràmetres del dispositiu electrònic com el nivell de sensibilitat del sensor d'humitat —transmetent-li una ordre de canvi de sensibilitat a través d'una comunicació Bluetooth— o la llista de terminals que aquest té emmagatzemat —transmetent-li una ordre d'inici de nova búsqueda de terminals, també a través de Bluetooth—. De manera extraordinària, també ha de ser capaç de transmetre l'historial d'alarmes enregistrades a l'aplicació de seguiment *PPStop* —que s'executa en un ordinador personal— també utilitzant una connexió Bluetooth.

Havent recordat totes aquestes funcionalitats que ha de tenir l'aplicació i recordant que en la seva fase de disseny vista en el capítol 2 es van prendre tot un seguit de decisions, aquest apartat es centra únicament en l'anàlisi de l'estructura de l'aplicació i el funcionament intern. Per a veure el funcionament de l'aplicació a nivell d'usuari s'ha elaborat una guia de l'usuari que es pot consultar a l'annex C.4.

### Estructura de paquets i classes

Internament, l'aplicació s'ha dividit en diferents paquets de classes, cadascun d'ells amb l'objectiu de cobrir unes determinades necessitats. Així ho mostra la taula 4.3.

Paquet	Classes	Funcionalitats que cobreix
<i>Audio</i>	Alarma, Audio	Conté les classes per a la creació, gestió i reproducció d'arxius d'àudio
<i>Bluetooth</i>	Connexió, Servei	Conté les classes per a crear el servei Bluetooth i gestionar tota comunicació amb l'aplicació
<i>Constants</i>	CONST, LANG	Conté les classes de constants numèriques i de text
<i>Dades</i>	BD, Historial, Config, Model, RMS	Conté les classes per a gestionar i emmagatzemar totes les dades de l'aplicació
<i>Main</i>	PPStop	Conté la classe principal de l'aplicació
<i>Utils</i>	Grafics, Informe, Utils, UtilsVista	Conté les classes amb funcions auxiliars de certa utilitat
<i>Vista</i>	FormAlarma, FormConfig, FormHistorial, FormMenu, FormResum, FormSplash, Vista	Conté les classes per a generar i gestionar la GUI (Graphic User Interface) de l'aplicació

Taula 4.3: Paquets de classes de l'aplicació del terminal mòbil



### 4.3. DESENVOLUPAMENT DE L'APLICACIÓ

---

Cada paquet, com es pot veure, conté una sèrie de classes i/o interfícies, i aquestes s'encarreguen d'una funció específica dins del paquet i, en general, dins de l'aplicació. A continuació es descriu la funcionalitat de cada classe dins de l'aplicació.

- **Audio.Alarma:** Classe que crea un *player* per a la reproducció de l'alarma *PPStop*.
- **Audio.Audio:** Classe que crea un *player* per a la reproducció d'arxius d'àudio.
- **Bluetooth.Connexió:** Classe que obre els *streams* d'e I/O per a la transmissió i recepció de dades.
- **Bluetooth.Servei:** Classe que crea registra el servei Bluetooth *PPStop* i tracta les connexions rebudes.
- **Constants.CONST:** Interfície que conté totes les constants numèriques utilitzades per l'aplicació.
- **Constants.LANG:** Interfície que conté totes les constants d'idioma utilitzades per l'aplicació.
- **Dades.BD:** Classe que actua de gestor de Base de Dades de l'aplicació.
- **Dades.Historial:** Classe representativa de l'Historial d'Alarmes.
- **Dades.Config:** Classe representativa de la configuració de l'aplicació.
- **Dades.Model:** Classe que conté totes les dades de l'aplicació.
- **Dades.RMS:** Classe que gestiona la *RMS* (Record Management System).
- **Main.PPStop:** Classe principal de l'aplicació *PPStop*.
- **Utils.Grafics:** Classe amb funcions útils d'àmbit general.
- **Utils.Informe:** Classe amb funcions útils en l'àmbit de la creació de la *GUI* (Graphic User Interface).
- **Utils.Utils:** Classe amb funcions útils d'àmbit general.
- **Utils.UtilsVista:** Classe amb funcions útils en l'àmbit de la creació de la *GUI* (Graphic User Interface).
- **Vista.FormAlarma:** Classe que defineix els elements gràfics del formulari d'Alarma.
- **Vista.FormConfig:** Classe que defineix els elements gràfics del formulari de Configuració.
- **Vista.FormHistoria:** Classe que defineix els elements gràfics del formulari d'Historial d'Alarmes.
- **Vista.FormMenu:** Classe que defineix els elements gràfics del formulari de Menú Principal.
- **Vista.FormResum:** Classe que defineix els elements gràfics del formulari de Resum.
- **Vista.FormSplash:** Classe que defineix els elements gràfics del formulari de Splash.

- **Vista.Vista:** Classe que conté tota la gestió de la *GUI* (Graphic User Interface).

Cada classe té uns determinats mètodes i atributs, que es poden consultar en el *JavaDoc* de l'aplicació que s'adjunta en l'annex C.2. Tanmateix, també es pot consultar el codi *JAVA* de l'aplicació —veure annex C.3— on es mostra com s'ha programat cada classe. Una vegada descrites les funcionalitats de les classes i interfícies, es passa a descriure com interaccionen entre elles seguint el patró *MVC* per aconseguir que l'aplicació funcioni correctament.

## Funcionament de l'aplicació

La interacció entre les diferents classes de l'aplicació depèn directament de l'aplicació que s'ha fet del patró *MVC*. La figura 4.14 mostra una estructura simplificada de les classes de l'aplicació i la intercomunicació entre elles. Es pot consultar el diagrama de classes complet en l'annex C.1. Començarem identificant-ne els elements. Per una banda es té l'aplicació del terminal mòbil —aplicació de control— i per l'altra les dos aplicacions que formen part del sistema global i amb les quals l'aplicació del terminal mòbil s'hi intercanvia informació.

Dins de l'aplicació de control i seguint el patró *MVC*, hi ha el model de dades —format per la classe *Model* i les classes de les quals en depèn—, la vista o interfície gràfica de l'usuari —formada per la classe *Vista* i els diferents formularis que conformen les pantalles de les diferents seccions— i el controlador —en aquest cas format per la classe principal *PPStop* i la classe que s'encarrega de gestionar el servei Bluetooth *PPStop* i tractar les connexions rebudes—. No obstant, també hi ha altres classes que tenen altres objectius i no s'inclouen dins de la funcionalitat de Model, Vista o Controlador. Aquest és el cas de les classes que s'encarreguen de generar l'alarma sonora i les que s'encarreguen de donar suport addicional en general com ara les interfícies *CONST* i *LANG* i les classes que proporcionen funcions d'utilitat tant a nivell general com a nivell de la vista.

Quan l'aplicació de control s'executa en el terminal mòbil, se'n crida la seva classe principal —la classe *PPStop*—, que és l'encarregada de posar tota l'aplicació en funcionament. El primer que fa és crear un objecte de la classe *Vista* —que mitjançant *LWUIT* genera la *SplashScreen* i conseqüentment, el menú principal—, crear un objecte de la classe *Model* —que el primer que fa és obtenir els objectes *Config* i *Historial* a través del *RMS*, recuperant així totes les dades guardades de l'aplicació—, crear un objecte de la classe *Alarma* —que mitjançant la classe *Audio* i les *MMAPI* posa a punt l'alarma sonora— i finalment crea un *thread* de la classe *Servei* —que gràcies a les *JABWT* registra el servei *PPStop* a la *SDDb* i es queda a l'espera de rebre connexions Bluetooth—. Aleshores, l'aplicació arriba a un punt en què pot reaccionar a dos events:

1. **Interacció per part de l'usuari:** En aquest cas, l'usuari navega per la interfície gràfica —el menú principal— i es van generant events. Aquests, són captats pel controlador —la classe principal *PPStop*— que en funció de l'event decideix l'ac-

### 4.3. DESENVOLUPAMENT DE L'APLICACIÓ

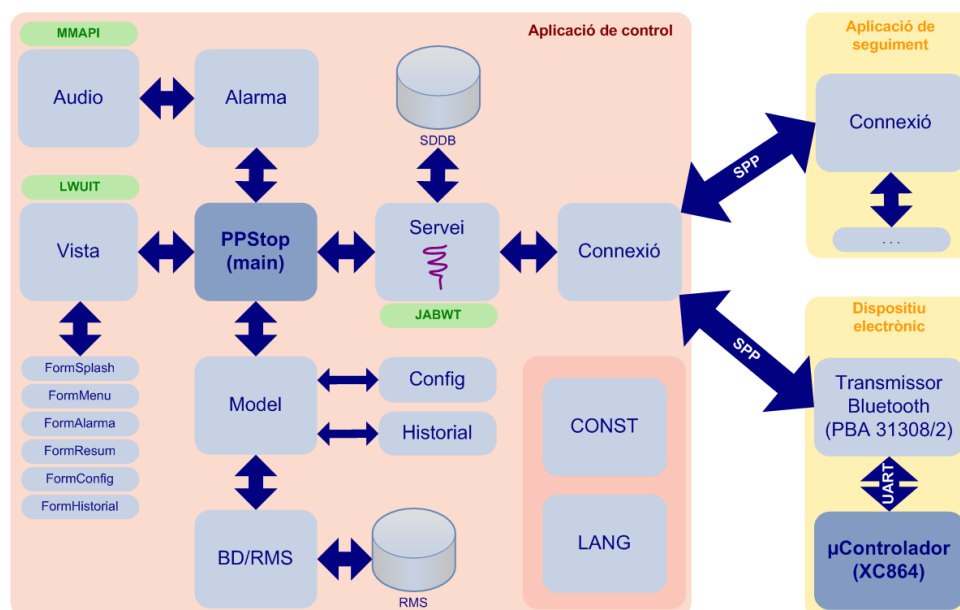


Figura 4.14: Estructura simplificada de l'aplicació de control

ció a fer. Suposem, per posar un cas concret, que l'usuari decideix seleccionar l'element del menú *Configuració* per tal de configurar l'aplicació. En aquest cas, el controlador decideix notificar a la vista —classe *Vista*— que ha de carregar la interfície gràfica corresponent a la classe *formConfig*, per tant aquesta es carrega i mostra les dades actualitzades proporcionades pel model. Suposem que, tot seguit, l'usuari canvia l'idioma de l'aplicació a través de la pantalla de configuració. Aleshores es genera un event al controlador —la classe *PPStop*—, aquesta decideix prendre l'acció de modificar les dades del model —la variable d'idioma de l'aplicació—, i aquesta genera la reacció d'actualitzar la vista i guardar el nou model modificat en el *RMS*.

2. **Recepció d'una connexió Bluetooth:** En aquest cas, el *thread Servei* tracta la connexió —event— i en funció de l'event generat —l'ordre rebuda via Bluetooth— pren una decisió. Suposem inicialment que l'ordre rebuda és la de verificar el servei; aleshores consulta el codi *PPStop* emmagatzemat en el model i suposant que són equivalents, actualitza l'estat de l'aplicació —a estat associat— canviant el valor de la variable corresponent en el model. El model en conseqüència, actua sobre la vista per mostrar aquest canvi —suposant per exemple que l'usuari està a la pantalla de resum de l'aplicació—. Suposem ara que en comptes de rebre l'ordre de verificar, rep una ordre d'alarmar. Aleshores, la classe *Servei* —que actua de controlador— consulta al model el *flag* d'aplicació. Suposem que el flag és el d'alarmar; aleshores el controlador produeix l'acció d'activar l'alarma sobre l'objecte de la classe *Alarma* i es genera l'alarma sonora.

Vist aquest anàlisi, se'n podrien fer de similars per a la resta de casos i possibilitats

de l'aplicació, però la manera d'operar és sempre la mateixa: event (per part de l'usuari o una connexió Bluetooth entrant) - acció (per part del controlador; sobre el model, la vista o l'alarma) - reacció (sobre la vista i sobre *RMS*). Queda ara més clar doncs que el model fa el paper d'*observable* i el controlador —i en certa manera també la vista— fan el paper d'*observador*.

Amb tot això, queda descrit el funcionament intern de l'aplicació. Per acabar, a la figura 4.15 es mostren tres captures de l'interfície gràfica de l'aplicació desenvolupada amb les llibreries *LWUIT*. En concret, es mostra la *Splash Screen* —pantalla que es mostra en posar en funcionament l'aplicació—, el menú principal —pantalla que apareix després de la *Splash Screen* i que permet accedir a les diferents parts de l'aplicació— i la pantalla de resum de dades —pantalla que es mostra en accedir a l'opció *Resum de Dades* i que mostra un resum de l'estat del sistema *PPStop*—.

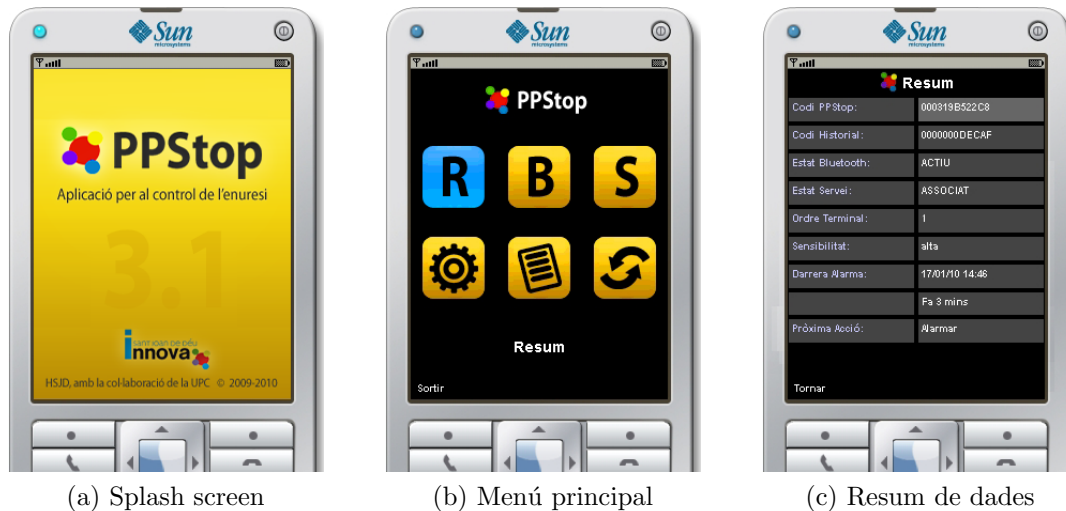


Figura 4.15: Captures de l'aplicació PPStop

També s'ha realitzat un estudi dels models de terminals mòbils que els principals fabricants tenen en mercat i que són compatibles amb l'aplicació segons les especificacions i requisits analitzats en els apartats 4.1 i 4.2 d'aquest mateix capítol. S'ha elaborat una llista dels terminals compatibles, que es pot consultar en l'annex C.5.

# Capítol 5

## Dispositiu electrònic

### 5.1 Disseny de la Unitat de Control (UC)

Una vegada preses les decisions de disseny respecte els components electrònics que formen el dispositiu electrònic *PPStop* i que s'han vist en el capítol 2, en aquesta secció s'analitza el disseny de la unitat de control del microcontrolador *XC684C* que s'encarrega de transmetre les comandes necessàries al transmissor Bluetooth *PBA31308/2* d'Infineon, recollir-ne les dades i respostes que aquest genera i actuar en conseqüència segons el protocol establert en el capítol 3.

#### 5.1.1 Comandes SPP-AT d'Infineon

El transmissor Bluetooth *PBA31308/2*, les característiques del qual es poden consultar a l'annex B.1, és un dispositiu electrònic que implementa la pila de protocols Bluetooth i que pot ser controlat a través d'unes comandes *AT* específiques del seu fabricant: *Infineon*. Aquestes comandes permeten resetejar el transmissor, configurar-lo, buscar dispositius Bluetooth i els seus serveis en el seu rang de cobertura, establir una connexió amb un altre dispositiu Bluetooth, transmetre dades, etc. En definitiva, que hi ha definides les suficients comandes *AT* com per a donar-li les ordres necessàries per a complir el protocol que serveix per comunicar-se amb l'aplicació de control *PPStop* del terminal mòbil.

Tot seguit es presenten les comandes *AT* d'Infineon i les seves respostes que són d'interès per al sistema *PPStop*, segons especifica la descripció de software d'Infineon —*eBMU SPP-AT*— i les quals s'adjunten en l'annex B.4. Les comandes es transmeten al transmissor a través d'una comunicació sèrie per la *UART*. La taula 5.1 mostra la llista de comandes *AT*, mentre que la taula 5.2 mostra les possibles respostes que pot generar el *PBA31308/2*. Les comandes *AT* segueixen un format força estàndard: *AT+[codi\_ordre]=[paràmetres]*. En canvi, hi ha dos tipus generals de resposta; les de confirmació a comandes *AT* transmeses al transmissor Bluetooth i les que genera direc-

Comanda AT	Descripció
<i>AT+JRES</i>	Reseteja el dispositiu bluetooth
<i>AT+JSEC=1,1,1,04,1111</i>	Estableix el mode de seguretat: Mode 1; Informa sobre la clau d'enllaç; PIN variable; PIN de longitud 4; PIN=1111
<i>AT+JSLN=06,PPSTOP</i>	Estableix el nom del dispositiu bluetooth: Nom de longitud 6; NOM=PPSTOP
<i>AT+JDDS=0</i>	Inicia la búsqueda de dispositius (DDP): Sense límit de resultats
<i>AT+JSDS=@mac,1101</i>	Inicia la búsqueda de serveis en un dispositiu (SDP): Adreça mac destí; UUID=1101 (SPP)
<i>AT+JCCR=@mac,#ch</i>	Crea una petició de connexió: Adreça mac destí; Canal del servei
<i>AT+JSDA=longitud,dades</i>	Envia dades al dispositiu remot: Longitud de les dades (bytes); Dades a enviar

Taula 5.1: Descripció de les comandes AT utilitzades

tament el transmissor degut a l'execució de les ordres que se l'hi envien per la *UART*. Les respostes de confirmació de les comandes *AT* acostumen a ser estàndards —*ROK*, *OK*, *ERR=-[valor]*— mentre que les generades per la propia activitat Bluetooth del transmissor tenen un format *+ [codi\_resposta]=[paràmetres]*.

### 5.1.2 Diagrama d'estats

Una vegada conegudes les comandes *AT* i les seves possibles respostes, es pot generar una transformació del diagrama de flux del protocol general vist en el capítol 3 en un diagrama d'estats que representi una màquina d'estats, on els estats de la màquina es corresponguin amb els possibles estats del protocol dissenyat i on les transicions entre els estats estiguin definides per les comandes *AT* transmeses al *PBA31308/2*, per les respostes procedents d'aquest i per la informació dels terminals *PPStop* associats al servei.

El diagrama d'estats de la unitat de control per al microcontrolador, resultant de fer aquesta transformació, és el mostrat en la figura 5.1. Les etiquetes colorejades amb color blau defineixen les comandes *AT* transmeses per la *UART* al transmissor Bluetooth en cada estat, les colorejades amb color verd defineixen les accions a realitzar en aquell estat i les colorejades amb groc matisen el significat d'algunes de les respostes obtingudes pel transmissor Bluetooth o bé el tipus d'event a captar pel sistema.

Resposta	Descripció
<i>ROK</i>	Reset realitzat correctament
<i>OK</i>	Comanda executada correctament
<i>ERR=-codi</i>	Error en l'execució de la comanda
<i>+RDDSRES=@mac,nom,tipus</i>	Nou dispositiu trobat durant el DDP: Adreça mac remota; Nom dispositiu remot; Tipus de dispositiu remot
<i>+RDDSCNF=estat</i>	La búsqueda de dispositius (DDP) ha finalitzat
<i>+RSDSRES=nom,canal</i>	Nou servei trobat durant el SDP: Nom del servei remot; Canal d'operació del servei remot
<i>+RSDSCNF=estat</i>	La búsqueda de serveis (SDP) ha finalitzat
<i>+RDAI=longitud,dades</i>	Dades rebudes: Longitud de les dades (bytes); Dades enviades
<i>+RDII</i>	El dispositiu remot s'ha desconnectat
<i>+RCCRCNF=MTU,estat</i>	Resposta a una petició de connexió: Tamany de la MTU; Estat de la connexió (0: OK, 1: KO)

Taula 5.2: Descripció de possibles respostes a les comandes AT

## 5.2 Desenvolupament del simulador

Prèviament a la programació de la unitat de control del microcontrolador, es va creure adequat la seva simulació per comprobar que el protocol estava ben dissenyat, que actuava tal i com s'esperava en la seva fase de disseny i per fer-ne els ajustos pertinents. Per a això es va programar un simulador utilitzant el llenguatge de programació *J2SE* —la versió estàndard de *JAVA*—, que permetés definir la lògica de la màquina d'estats vista anteriorment i comunicar-se amb una versió de testeig del transmissor Bluetooth.

El simulador va ser programat per a executar-se en un ordinador personal de sobretaula normal i corrent, i la comunicació amb el transmissor ha estat possible gràcies a la Java Communications API (COMMAPI), una *API* addicional per a *JAVA* que permet obrir una comunicació sèrie amb un dispositiu extern —en aquest cas amb el *Development Kit* del PBA31308/2— i poder així transmetre-li les comandes *AT* necessàries i llegir-ne les respostes obtingudes.

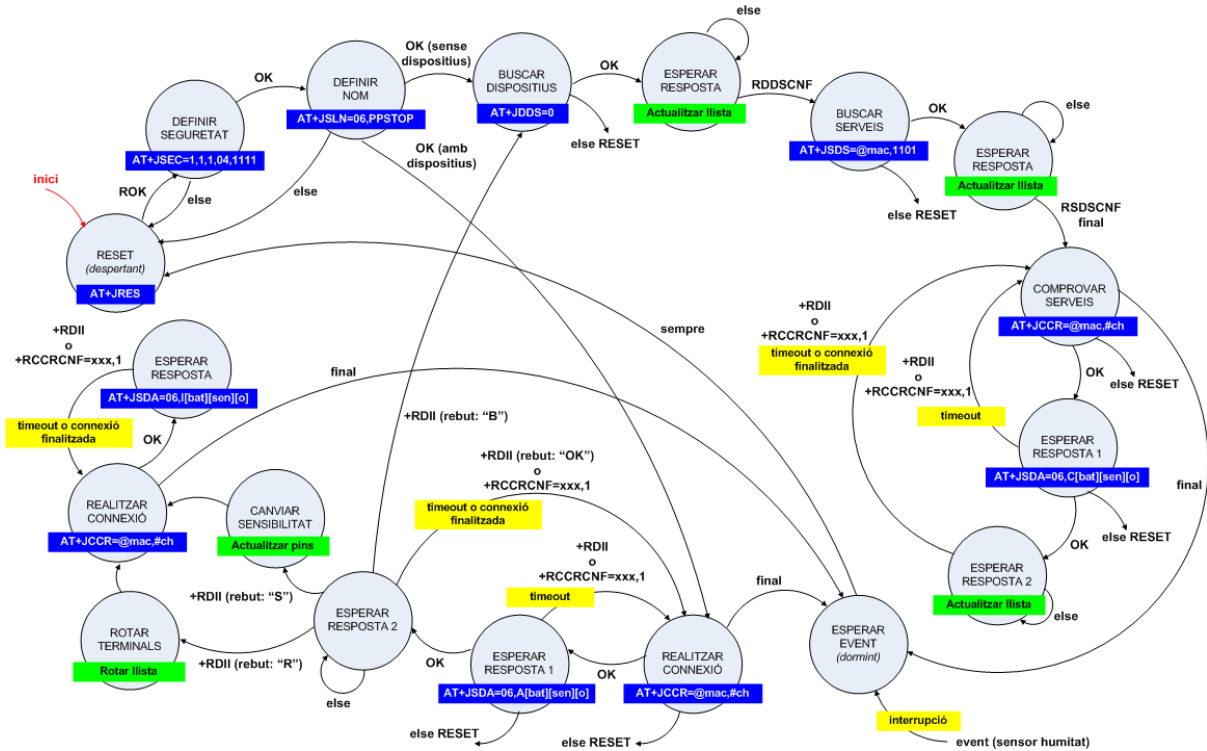


Figura 5.1: Diagrama d'estats de la UC

### 5.2.1 Java Communications API (COMMAPI)

L'*API* de comunicacions de *JAVA*[62] —també coneguda amb el nom de *javax.comm*, veure figura 5.2— és una extensió que facilita una plataforma independent de desenvolupament d'aplicacions per a tecnologies com ara Smarts Cards, TPV's, modems, equipament de rebòtica, etc. Poporciona a les aplicacions l'accés al hardware *RS-232* —ports sèrie— i accés limitat a l'estàndard *IEEE-1284* —ports paral·lels—, ambdós en mode *SPP*.

Algunes funcionalitats sèrie que ofereix l'*API* són:

- Enumeració de ports —mapeig de ports configurable per a administrador i usuari—
- Configuració del port —velocitat de transmissió, bits de dades / aturada i paritat—
- Accés a l'estàndard *EIA-232*, senyals *DTR*, *CD*, *CTS*, *RTS*.
- Transferència de dades sobre port *RS-232*.
- Opcions de control de flux software i hardware.
- Control de lllindar del buffer de lectura.
- Opció d'event asíncron per a notificació de: dades disponibles en el port *RS-232*, canvis de la línia de nivell del port hardware, canvi del propietari del port dins d'una sola *JVM*.



## 5.2. DESENVOLUPAMENT DEL SIMULADOR

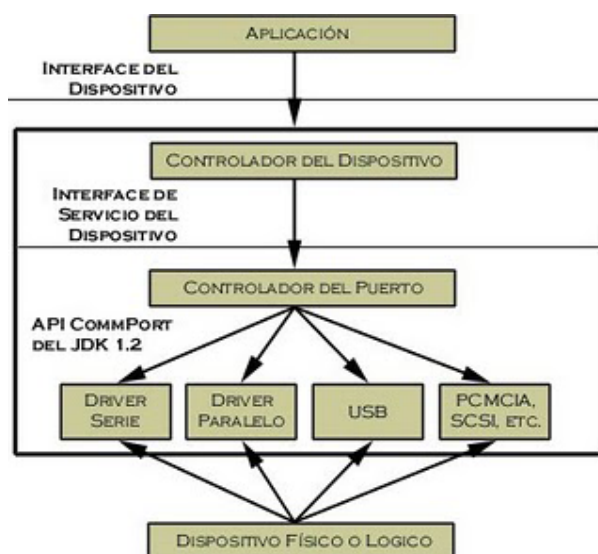


Figura 5.2: Visió global de la Java COMMAPI

En l'aplicació *JAVA* del simulador, s'enumeren i configuren adequadament els ports sèrie amb els paràmetres requerits pel Development Kit del transmissor Bluetooth[63]:

- **Velocitat de transmissió:** 115200 bauds.
- **Bits de dades:** 8 bits.
- **Bits d'aturada:** 1 bit.
- **Paritat:** sense paritat.

A més a més s'utilitza l'event *SerialPortEvent.DATA\_AVAILABLE* per a detectar quan s'han rebut noves dades pel port sèrie, procedir a la seva lectura i així posteriorment poder actuar en conseqüència segons la màquina d'estats definida.

### 5.2.2 Development Kit per a Infineon PBA31308/2

El fabricant *Infineon* ofereix una versió de testeig del transmissor Bluetooth *PBA31308/2* amb les mateixes funcionalitats, amb connectivitat *USB* i un convertidor intern *USB-UART* per a la transmissió de la informació i les comandes *AT* —veure en la figura 5.3—.

Aquest dispositiu, tot i ser una versió per a *developers*, té les mateixes característiques que el *PBA31308/2* comentat en el capítol 2 —Bluetooth 2.0 + *EDR*, una única connexió simultània, etc.— i és el dispositiu que s'ha fet servir per a fer la simulació de la unitat de control. En les fotografies de la figura es pot apreciar el connector *USB* per a establir la connexió sèrie amb el *PC*, i en l'estructura interna del dispositiu es pot apreciar com a sota de la capa de les comandes *AT* el dispositiu hi té implementada tota la pila de protocols Bluetooth comentada en el capítol 3.

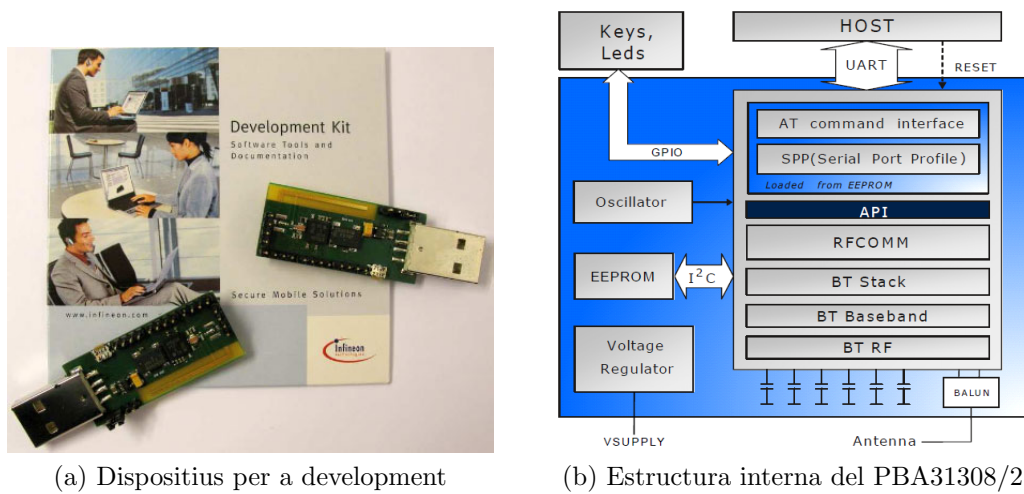


Figura 5.3: Development Kit per a Infineon PBA31308/2

### 5.2.3 Estructura i funcionament

Fins ara s'ha descrit la màquina d'estats de la unitat de control i el component que ha facilitat *Infineon* per a possibilitar la simulació del comportament del prototip abans de fabricar-lo. Tot seguit passa a analitzar-se com funciona el simulador a nivell intern, tal com es pot veure a la figura 5.4. Se'n pot observar la seva estructura de classes en l'annex A.1 i veure la funcionalitat detallada de cada classe en el JavaDoc a l'annex A.2. Per a consultar el codi *JAVA* d'alguna de les classes que formen el simulador, aquest es pot consultar a l'annex A.3.

Quan s'executa el simulador, s'instancia la classe *Simulador* —que és la principal del Simulador— i aquest crea un objecte de la classe *ComunicacioSerie* —que mitjançant *COMMAPI* configura el port sèrie on està connectat el transmissor Bluetooth de testeig i n'obre una connexió sèrie—, crea la *GUI* a través del Event Dispatcher Thread (EDT), crea un objecte del model de dades —que conté totes les dades del simulador: dispositius Bluetooth en memòria, nombre d'alarmes enregistrades, connexions establertes, etc.— i finalment crea un objecte de la classe *UnitatControl* que conté la lògica de la màquina d'estats dissenyada en l'apartat 5.1. En aquest moment, el simulador ja està llest per a funcionar i pot iniciar el protocol emulant l'acció de mullar el sensor d'humitat del dispositiu a través d'un botó habilitat per a tal efecte en la interfície gràfica d'usuari del simulador.

En aquesta distribució de classes s'ha fet ús del patró de programació *MVC* de la mateixa manera que s'ha fet en l'aplicació del terminal mòbil. La classe *GUI* clarament fa el paper de Vista, la classe *Model* de model, i la classe principal —*Simulador*— i la *Comunicació Serie* actuen de controladors —el primer a l'espera d'un event en la *GUI* per part de l'usuari del simulador i el segon a l'espera d'un event d'arribada de dades per part del transmissor Bluetooth—. Es poden veure un parell de captures de la pantalla

## 5.2. DESENVOLUPAMENT DEL SIMULADOR

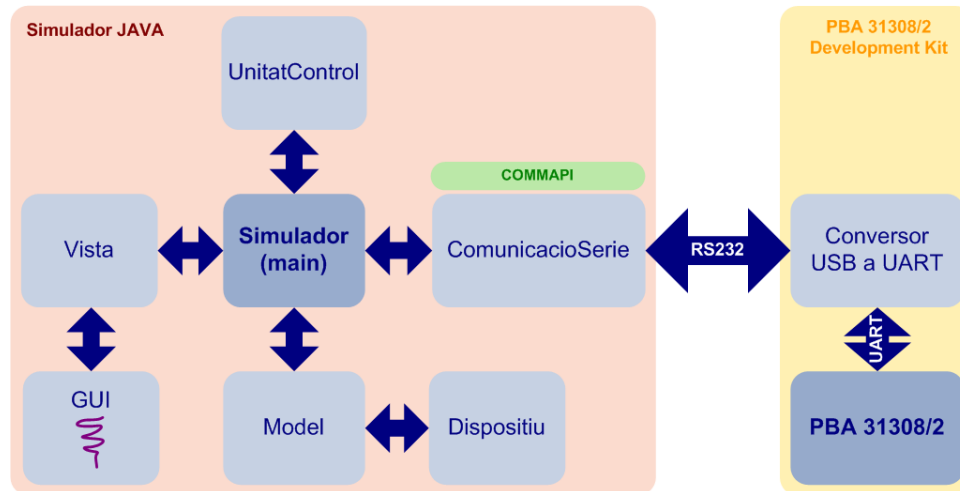


Figura 5.4: Estructura simplificada del simulador

del simulador a la figura 5.5

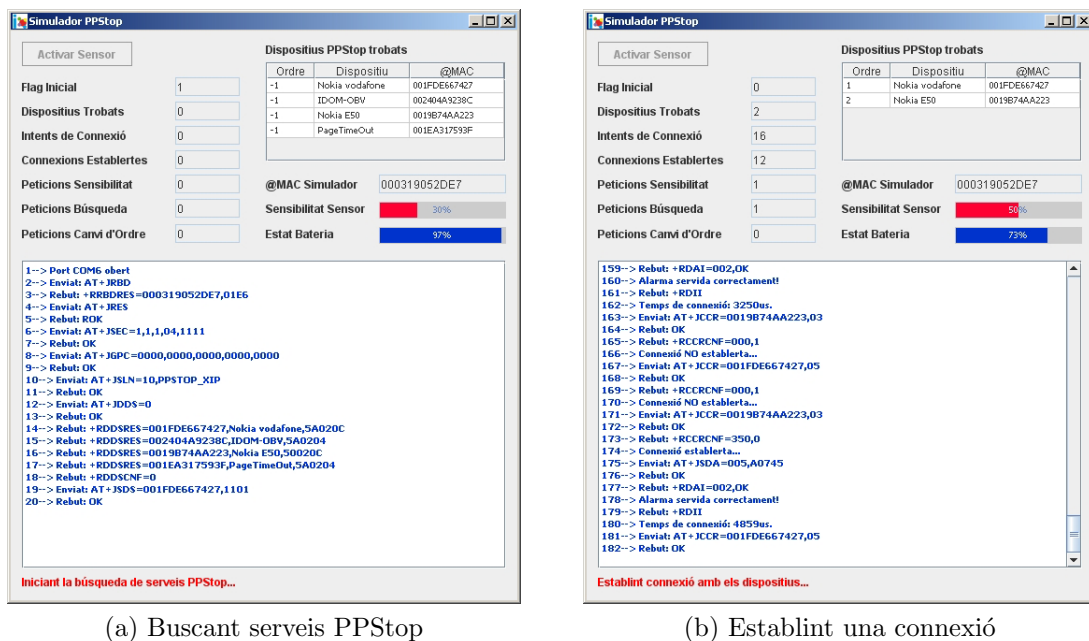


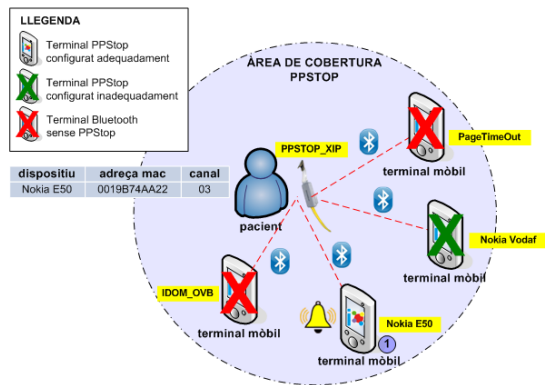
Figura 5.5: Captures del simulador

### 5.2.4 Traces de la UC

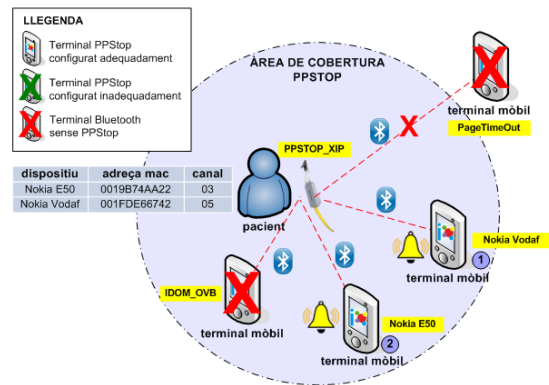
Les traces de la UC són els logs que s'obtenen a la finestra de log de la interfície gràfica del simulador i són molt importants perquè deixen constància de totes les comandes enviades per la UC, les respostes generades pel transmissor i les dades intercanviades amb el

terminal mòbil. Addicionalment, es mostren estampes de temps que permeten analitzar certs aspectes de temps del protocol, com ara el temps d'establiment de connexió o el temps complet de generació d'una alarma. L'anàlisi d'aquestes traces ha ajudat a depurar i aconseguir una programació del protocol que contemplés tots els casos possibles i que sempre actués correctament, sense quedar-se penjat en cap punt de la màquina d'estats. Això ha resultat vital de cara a la programació de la unitat de control del microcontrolador, que com es veurà més endavant, careix de mètodes eficients de depuració.

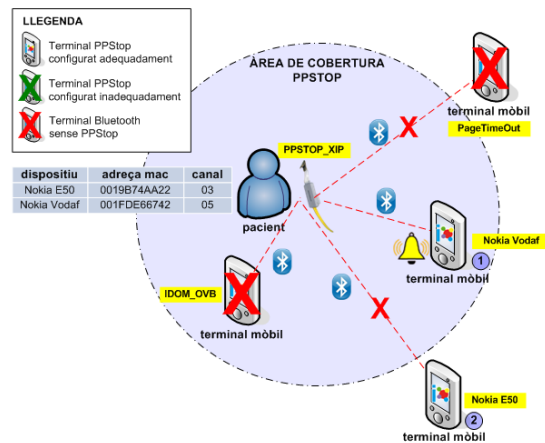
Tot seguit es mostren les traces obtingudes durant algunes simulacions de situacions habituals en les que es pot arribar a trobar el sistema. Aquests escenaris es mostren a la figura 5.6, on se n'il·lustren 4 casos que es corresponen a les traces obtingudes i exposades en la taula 5.3. Cal remarcar que aquestes simulacions són prèvies a la versió final del protocol—veure capítol 3— i conté algunes petites diferències: no existeix l'ordre de *rotar terminals (R)* ni la d'*actualitzar informació (I)*, i no es transmet l'ordre del terminal. A més a més, el nivell de sensibilitat té 5 nivells (1-5) en comptes de 2 únics nivells (0/1).



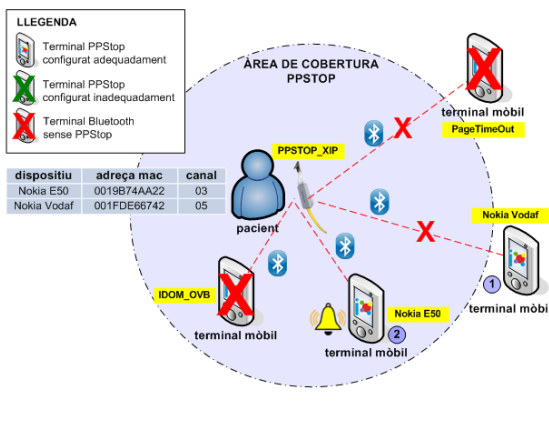
(a) Escenari simulador A



(b) Escenari simulador B



(c) Escenari simulador C



(d) Escenari simulador D

Figura 5.6: Escenaris per al testeig de la UC mitjançant el simulador

## 5.2. DESENVOLUPAMENT DEL SIMULADOR

Taula 5.3: Traces de la UC obtingudes amb el simulador

(a) Configuració inicial	
#	T Traça
1	- Port COM6 obert
2	- Iniciant configuració
3	W AT+JRBD
4	R +RRBDRES=000319052DE7,01E6
5	W AT+JRES
6	R ROK
7	W AT+JSEC=1,1,1,04,1111
8	R OK
9	W AT+JGPC=0000,0000,0000,0000,0000
10	R OK
11	W AT+JSLN=10,PPSTOP_XIP
12	R OK
13	- Configuració finalitzada
(b) Búsqueda de terminals	
#	T Traça
14	- Iniciant búsqueda de terminals
15	W AT+JDDS=0
16	R OK
17	R +RDDSRES=001FDE667427,Nokia vodaf,5A020C
18	R +RDDSRES=002404A9238C,IDOM-OBV,5A0204
19	R +RDDSRES=0019B74AA223,Nokia E50,50020C
20	R +RDDSRES=001EA317593F,PageTimeOut,5A0204
21	R +RDDSCNF=0
22	- Búsqueda de terminals finalitzada
(c) Búsqueda de serveis	
#	T Traça
23	- Iniciant búsqueda de serveis
24	W AT+JSDS=001FDE667427,1101
25	R OK
26	R +RSDSRES=PPStop,05
27	R +RSDSCNF=0
28	W AT+JSDS=002404A9238C,1101
29	R OK
30	R +RSDSRES=Nokia PC Suite,15
31	R +RSDSRES=COM 1,03
32	R +RSDSCNF=0
33	W AT+JSDS=0019B74AA223,1101
34	R OK
35	R +RSDSRES=PPStop,03
36	R +RSDSCNF=0
37	W AT+JSDS=001EA317593F,1101
38	R OK
39	R +RSDSCNF=4
40	- Búsqueda de serveis finalitzada
(d) Verificació de serveis	
#	T Traça
41	- Iniciant verificació de serveis
42	W AT+JCCR=001FDE667427,05
43	R OK
44	R +RCCRCNF=350,0
45	- Connexió establerta...
46	W AT+JSDA=005,C0933
47	R OK
48	R +RDII
49	- El servei s'ha verificat i ha sigut REBUTJAT...
50	W AT+JCCR=0019B74AA223,03
51	R OK
52	R +RCCRCNF=350,0
53	- Connexió establerta...
54	W AT+JSDA=005,C0923
55	R OK
56	R +RDAI=002,OK
57	- El servei s'ha verificat i ha sigut ACCEPTAT...
58	R +RDII
59	- Verificació de serveis finalitzada
(e) Generació d'alarma (2 consecutives) (terminal en cobertura)	
#	T Traça
60	- Iniciant generació d'alarma
61	W AT+JCCR=0019B74AA223,03
62	R OK
63	R +RCCRCNF=350,0
64	- Connexió establerta...
65	W AT+JSDA=005,A0913
66	R OK
67	R +RDAI=002,OK
68	- Alarma servida correctament!
69	R +RDII
70	- Temps de connexió: 968ms.
71	W AT+JCCR=0019B74AA223,03
72	R OK
73	R +RCCRCNF=350,0
74	- Connexió establerta...
75	W AT+JSDA=005,A0903
76	R OK
77	R +RDAI=002,OK
78	- Alarma servida correctament!
79	R +RDII
80	- Temps de connexió: 1578ms.
81	- Generació d'alarma finalitzada
(f) Generació d'alarma (terminal fora de cobertura)	
#	T Traça
82	- Iniciant generació d'alarma
83	W AT+JCCR=0019B74AA223,03
84	R OK
85	R +RCCRCNF=000,1
86	- Connexió NO establerta...
87	- Generació d'alarma finalitzada
(g) Canvi de sensibilitat	
#	T Traça
88	- Iniciant generació d'alarma
89	W AT+JCCR=0019B74AA223,03
90	R OK
91	R +RCCRCNF=350,0
92	- Connexió establerta...
93	W AT+JSDA=005,A0883
94	R OK
95	R +RDAI=002,S5
96	- Nova ordre: canvi de sensibilitat (nivell 5)

97 R +RDII  
 98 - Temps de connexió: 1625ms.  
 99 - Generació d'alarma finalitzada

(h) Nova búsqueda de terminals

#	T	Traça
100	-	Iniciant generació d'alarma
101	W	AT+JCCR=0019B74AA223,03
102	R	OK
103	R	+RCCRCNF=350,0
104	-	Connexió establerta...
105	W	AT+JSDA=005,A0875
106	R	OK
107	R	+RDAI=001,B
108	-	Nova ordre: búsqueda de dispositius
109	R	Rebut: +RDII
110	-	Temps de connexió: 1532ms.
111	-	Generació d'alarma finalitzada
112	W	AT+JDDS=0
113	R	OK
114	R	+RDDSRES=002404A9238C,IDOM-OBV,5A0204
115	R	+RDDSRES=001FDE667427,Nokia vodafone,5A020C
116	R	+RDDSRES=0019B74AA223,Nokia E50,50020C
117	R	+RDDSCNF=0
118	W	AT+JSDS=002404A9238C,1101
119	R	OK
120	R	+RSDSRES=Nokia PC Suite,15
121	R	+RSDSRES=COM 1,03
122	R	+RSDSCNF=0
123	W	AT+JSDS=001FDE667427,1101
124	R	OK
125	R	+RSDSRES=PPStop,03
126	R	+RSDSCNF=0
127	W	AT+JSDS=0019B74AA223,1101
128	R	OK
129	R	+RSDSRES=PPStop,03
130	R	+RSDSCNF=0
131	W	AT+JCCR=001FDE667427,05
132	R	OK
133	R	+RCCRCNF=350,0
134	-	Connexió establerta...
135	W	AT+JSDA=005,C0815
136	R	OK
137	R	+RDAI=002,OK
138	-	El servei s'ha verificat i ha sigut ACCEPTAT...
139	R	+RDII
140	W	AT+JCCR=0019B74AA223,03
141	R	OK
142	R	+RCCRCNF=350,0
143	-	Connexió establerta...
144	W	AT+JSDA=005,C0805
145	R	OK
146	R	+RDAI=002,OK
147	-	El servei s'ha verificat i ha sigut ACCEPTAT...
148	R	+RDII
149	-	Búsqueda de serveis finalitzada

(i) Generació d'alarma  
(2 terminals en cobertura)

#	T	Traça
150	-	Iniciant generació d'alarma

151	W	AT+JCCR=001FDE667427,05
152	R	OK
153	R	+RCCRCNF=350,0
154	-	Connexió establerta...
155	W	AT+JSDA=005,A0795
156	R	OK
157	R	+RDAI=002,OK
158	-	Alarma servida correctament!
159	R	+RDII
160	-	Temps de connexió: 4141ms.
161	W	AT+JCCR=0019B74AA223,03
162	R	OK
163	R	+RCCRCNF=350,0
164	-	Connexió establerta...
165	W	AT+JSDA=005,A0785
166	R	OK
167	R	+RDAI=002,OK
168	-	Alarma servida correctament!
169	R	+RDII
170	-	Temps de connexió: 7797ms.
171	-	Generació d'alarma finalitzada

(j) Generació d'alarma a 2 terminals  
(1r en cobertura, 2n fora de cobertura)

#	T	Traça
172	-	Iniciant generació d'alarma
173	W	AT+JCCR=001FDE667427,05
174	R	OK
175	R	+RCCRCNF=350,0
176	-	Connexió establerta...
177	W	AT+JSDA=005,A0775
178	R	OK
179	R	+RDAI=002,OK
180	-	Alarma servida correctament!
181	R	+RDII
182	-	Temps de connexió: 3250ms.
183	W	AT+JCCR=0019B74AA223,03
184	R	OK
185	R	+RCCRCNF=000,1
186	-	Connexió NO establerta...
187	-	Generació d'alarma finalitzada

(k) Generació d'alarma a 2 terminals  
(1r fora de cobertura, 2n en cobertura)

#	T	Traça
188	-	Iniciant generació d'alarma
189	W	AT+JCCR=001FDE667427,05
190	R	OK
191	R	+RCCRCNF=000,1
192	-	Connexió NO establerta...
193	W	AT+JCCR=0019B74AA223,03
194	R	OK
195	R	+RCCRCNF=350,0
196	-	Connexió establerta...
197	W	AT+JSDA=005,A0745
198	R	OK
199	R	+RDAI=002,OK
200	-	Alarma servida correctament!
201	R	+RDII
202	-	Temps de connexió: 4859ms.
203	-	Generació d'alarma finalitzada

### 5.3. PREPARACIÓ DEL DISPOSITIU

---

Per a aquesta taula, és té que:

- Les traces de (a),(b),(c),(d),(e) i (f) corresponen amb l'escenari simulador A
- Les traces de (h) i (i) corresponen amb l'escenari simulador B
- Les traces de (j) corresponen amb l'escenari simulador C
- Les traces de (k) corresponen amb l'escenari simulador D

## 5.3 Preparació del dispositiu

### 5.3.1 Estructura i components

Per a la implementació del dispositiu electrònic s'ha construït un xip de  $5.5cm \times 2.5cm$  que conté tots els elements especificats en la fase de disseny vista en el capítol 2, interconnectats adequadament tal i com es detalla en l'esquema de connexions de l'apèndix B.6. L'apèndix B.7 mostra els *layouts* utilitzats per a la fabricació del dispositiu, amb la disposició exacta de tots els components electrònics i de les diferents pistes d'interconnexió. Una versió simplificada és la que es mostra en la figura 5.7.

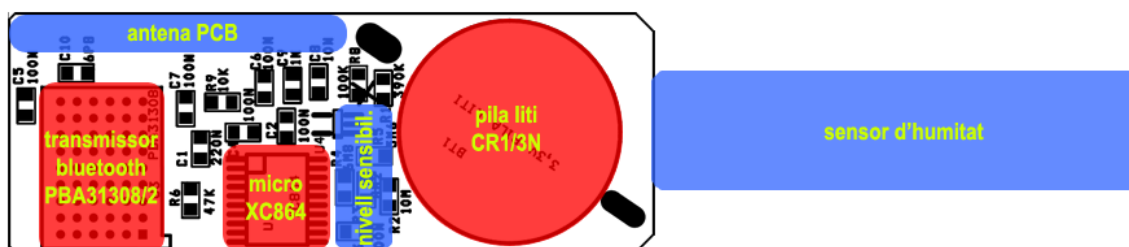


Figura 5.7: Estructura i elements del dispositiu electrònic

Els components inclosos en el xip són:

- **Pila de liti CR1/3N:** És la bateria de liti que alimenta a la resta de components del xip. És del fabricant *EEMB*, té unes dimensions molt reduïdes i ofereix una capacitat nominal de 160mAh. La resta de les seves característiques estan detallades en l'annex B.3.
- **Transmissor Bluetooth PBA31308/2:** És el mòdul transmissor Bluetooth del fabricant *Infineon*. En ser de classe 2 ofereix un rang de cobertura d'uns 10m, funciona amb Bluetooth 2.0 + *EDR* i implementa tota la pila de protocols Bluetooth. Com ja s'ha comentat, és comandable a través de comandes *AT* i l'intercanvi de



dades el fa a través de la *UART*. La resta de les seves característiques estan detallades en l'annex B.1[64] i la seva documentació tècnica referent a hardware i a software es pot consultar a [63] i a [65], respectivament.

- **Microcontrolador XC864:** És el microcontrolador del fabricant *Infineon* que actua d'unitat de control de la resta d'elements del xip, i que es mostra en la figura 5.8. És un model de 8 bits de la família XC800 d'Infineon[66], amb una capacitat de memòria de 4kB per a codi d'aplicació i amb una memòria ROM, RAM i capacitat de processament suficients per a la funcionalitat que se l'hi vol donar. Permet ser "adormit" per a estalviar energia i ser "despertat" a través de la recepció d'un event. La resta de les seves característiques estan detallades en l'annex B.2[67], i la seva documentació tècnica es pot consultar a [68].
- **Antena PCB:** Antena en el circuit imprès que actua d'antena del transmissor Bluetooth *PBA31308/2*.
- **Sensor d'humitat:** Sensor convencional de doble pista en forma de "tira", que amb la conducció entre les seves pistes mitjançant un conductor líquid provoca un pas de corrent a través seu i, per tant, una caiguda de tensió.

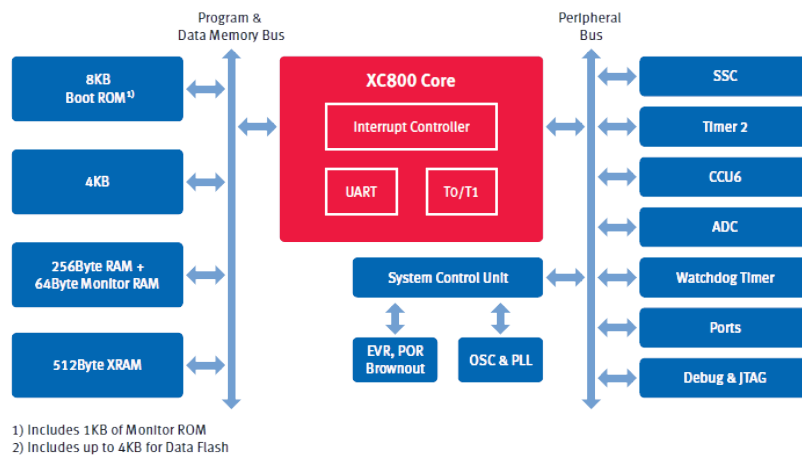


Figura 5.8: Estructura interna del XC864



(a) Infineon PBA31308/2



(b) Infineon XC864



(c) EEMB CR1/3N

Figura 5.9: Components del dispositiu electrònic





Figura 5.10: Easy Kit for Developing per a Infineon XC864

Pel que fa al mòdul marcat com a *nivell de sensibilitat*, no és cap component en si si no una agrupació de components electrònics que permeten definir el nivell de sensibilitat del sensor d'humitat i que es mostra en detall a l'apartat 5.3.3 d'aquest mateix capítol. La resta de components esmentats són els que es mostren en la figura 5.9.

#### 5.3.2 Programació del microcontrolador Infineon XC864

El procés de programació del microcontrolador inclou la programació de la unitat de control i la utilització de les eines necessàries per al testeig i la gravació del programa a la memòria del microcontrolador.

Per a la programació del microcontrolador s'ha transformat el codi de la unitat de control programada amb *JAVA* en el simulador a codi equivalent en llenguatge *C* que és el que admet el microcontrolador. Aquesta transformació s'ha realitzat fent ús de l'*Easy Kit for Developing per a l'Infineon XC864* [69] —veure figura 5.10—, que inclou un connector *USB* per a l'alimentació del microcontrolador, la transmissió de dades a través de la *UART* i la gravació del programa a la memòria flash del microcontrolador.

Per a la programació amb llenguatge *C* s'ha utilitzat el software *KEIL uVision* [70] que ofereix un compilador de *C* i un *debugger*, i el qual porta suport per a aquest model de microcontrolador. Per a la configuració del microcontrolador s'ha fet servir l'eina d'*Infineon* incorporada en l'*Easy Kit for Developing* anomenada *DAvE* [71], per a la gravació de l'aplicació s'ha fet servir l'aplicació *XC800\_FLOAT* present en aquest mateix kit, i finalment per a l'enviament de comandes *AT* i dades en general a través de la *UART* —fase de testeig— s'ha emprat l'aplicació *Docklight* [72].

Tot aquest procés s'ha realitzat seguint les indicacions del fabricant del microcontrolador [73] i seguint les pautes de programació que *KEIL* estableix per a aquest tipus de microcontrolador [74]. El codi del programa resultant gravat al microcontrolador té un pes de gairebé 4kB i es pot consultar en l'annex B.5.

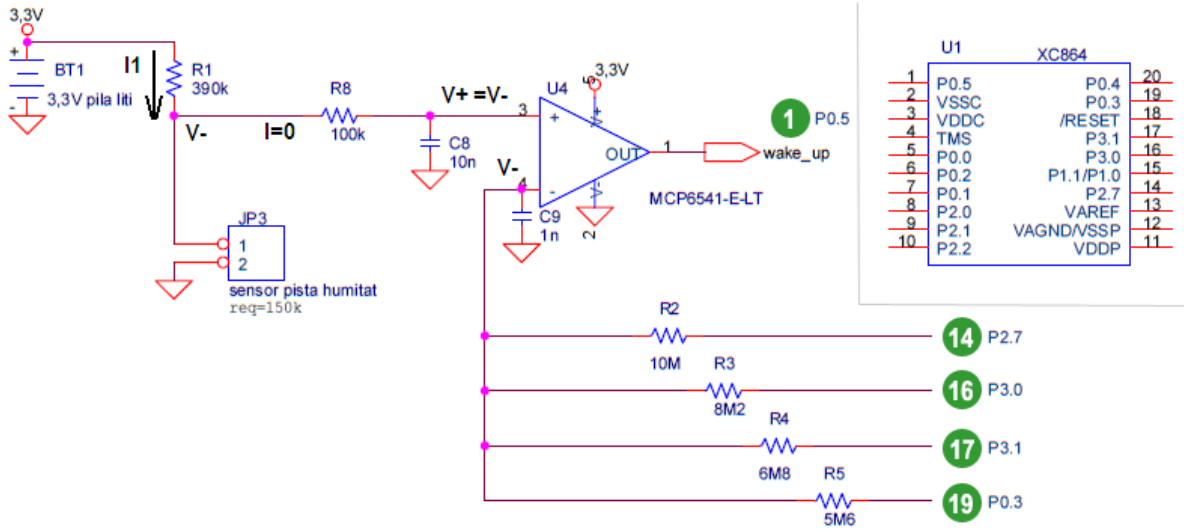


Figura 5.11: Circuit comparador per definir el nivell de sensibilitat del sensor d'humitat

### 5.3.3 Sensibilitat del sensor

El sensor d'humitat és el component a través del qual es genera l'event per a que el dispositiu iniciï un avís d'alarma als terminals *PPStop* que té guardats a memòria. Com es pot veure en l'esquema de connexions de la placa adjuntada en l'annexB.6 la pista del sensor està connectada a un comparador —entrada 3— que té establert una tensió de referència a l'altra entrada —entrada 4—. Aquesta tensió de referència és la que determina el nivell de sensibilitat del sensor, és a dir, a partir d'aquesta tensió el comparador activa el senyal de *wake up* —despertar— del microcontrolador.

La resistència  $R_8$  de valor  $100k\Omega$  actua com a protecció de l'entrada 3 del comparador ( $I_3 \approx 0$ ). Quan la pista del sensor ha de disparar l'alarma, aquesta té una resistència equivalent  $R_{eq} = 150k\Omega$ . Així doncs, la resistència  $R_1 = 390k\Omega$  (valor alt) s'encarrega de limitar-ne el seu consum.

El valor de la tensió de referència a l'entrada 4 ve determinat per les resistències  $R_2 = 10M\Omega$ ,  $R_3 = 8.2M\Omega$ ,  $R_4 = 6.7M\Omega$  i  $R_5 = 5.6M\Omega$ . Per establir diferents valors de tensió de referència es construeix un divisor de tensió amb les resistències que interessen, connectant-les a 3.3V o a massa segons convingui. Aquest darrer fet queda determinat pels corresponents valors dels ports *P3.0*, *P3.1* i *P0.1* del microcontrolador —un valor 1 en el port fa que la resistència estigui connectada a 3.3V, i un valor 0 fa que estigui connectada a massa— i el port *P2.7* que funciona diferent que la resta perquè no quedaven més ports de sortida disponibles en el microcontrolador. A diferència dels altres tres, aquest últim és un port d'entrada no configurable a sortida, per tant el que es fa és seleccionar el port en mode *pull-up* o en mode *pull-down*.

Connectar el *pull-up* i el *pull-down* equival a connectar un generador de corrent cons-

### 5.3. PREPARACIÓ DEL DISPOSITIU

---

tant d'intensitat limitada que, comparada amb la corrent que consumeix la resistència  $R_2 = 10M\Omega$  connectada equival a un port de sortida. Per tant, segons el *datasheet* del fabricant[68], es té que:

$$I_{pull-up} = I_{pull-down} = 5\mu A \quad (5.1)$$

Aleshores, buscant el cas de màxima corrent per a la resistència de  $10M\Omega$  —que es donarà quan el port 2.7 estigui seleccionat com a *pull-up* i els altres ports P3.0=0, P3.1=0 i P3.3=0— es té que:

$$I_{10M} = \frac{3.3V}{8.2M\Omega \parallel 6.8M\Omega \parallel 5.6M\Omega} = 0.27\mu A << 5\mu A \quad (5.2)$$

per tant, el *pull-up pull-down* equival a un port de sortida.

A partir d'aquí i sabent que el sensor d'humitat com hem dit anteriorment dispara el comparador quan  $V^+ > V^-$  —veure figura 5.11—, estudiant tots els possibles valors de la tensió  $V^-$ —tensió a l'entrada 4 del comparador— segons les combinacions possibles de  $R_2 = 10M\Omega$ ,  $R_3 = 8.2M\Omega$ ,  $R_4 = 6.7M\Omega$  i  $R_5 = 5.6M\Omega$ , és possible calcular els diferents valors d'intensitat que circularà pel sensor ja que  $V^+ = V^-$ . Posteriorment, això permet calcular la seva resistència equivalent en cada cas. La corrent que passa a través del sensor és la mateixa que passa a través de  $R_1$  ( $I_1$ ), i per tant s'ha de calcular a través de la diferència de tensió en els seus borns:

$$I_1 = \frac{3.3V - V^-}{390k\Omega} \quad (5.3)$$

Com hem dit, a partir de la corrent que passa pel sensor s'obté la seva resistència equivalent que dispara el sensor — $R_{sensor}$ —. Aquesta resistència inicialment és infinita ja que és la quantitat d'aigua —conductor— la que en determina el seu valor doncs a més aigua es té menys resistència.

Per determinar el tant per cent de sensibilitat, s'ha agafat un sensor i se n'ha calculat la seva resistència equivalent, obtenint un valor de  $390k\Omega$ . Aquest havia de ser un dels valors possibles, i equival a una sensibilitat del 8%. Per tant, per establir el % de sensibilitat en els demés casos s'agafa el valor més alt de les resistències possibles i se'n fa el tant per cent respecte d'aquest —a més resistència, més aigua serà necessària per a que es dispari el sensor—.

Tenint en compte l'expressió 5.3 i tot el raonament que s'acaba d'exposar, combinant totes les combinacions de les resistències connectades als ports del microcontrolador s'obté la següent taula de valors:

Resistències (ON ; OFF)	$3.3V - V^-$ (V)	$I_1$ ( $\mu A$ )	$R_{sensor}$ (k $\Omega$ )	Sensib. (%)	Niv	P2.7	P3.0	P3.1	P0.3
; $R_2, R_3, R_4, R_5$	0.00	8.46	0	<b>0</b>	-	-	-	-	-
$R_2$ ; $R_3, R_4, R_5$	0.60	6.92	87	<b>5</b>	-	1	0	0	0
$R_3$ ; $R_2, R_4, R_5$	0.73	6.58	112	<b>6</b>	-	0	1	0	0
$R_4$ ; $R_2, R_3, R_5$	0.88	6.19	143	<b>8</b>	-	0	0	1	0
$R_5$ ; $R_2, R_3, R_4$	1.07	5.70	189	<b>11</b>	<b>1</b>	0	0	0	1
$R_2, R_3$ ; $R_4, R_5$	1.34	5.02	266	<b>15</b>	<b>2</b>	1	1	0	0
$R_2, R_4$ ; $R_3, R_5$	1.49	4.64	321	<b>18</b>	<b>3</b>	1	0	1	0
$R_2, R_5$ ; $R_3, R_4$	1.68	4.16	404	<b>23</b>	-	1	0	0	1
$R_3, R_4$ ; $R_2, R_5$	1.62	4.30	377	<b>22</b>	-	0	1	1	0
$R_3, R_5$ ; $R_2, R_4$	1.81	3.82	474	<b>27</b>	<b>4</b>	0	1	0	1
$R_4, R_5$ ; $R_2, R_3$	1.96	3.43	572	<b>33</b>	<b>5</b>	0	0	1	1
$R_2, R_3, R_4$ ; $R_5$	2.22	2.76	806	<b>46</b>	-	1	1	1	0
$R_2, R_3, R_5$ ; $R_4$	2.41	2.27	1062	<b>61</b>	-	1	1	0	1
$R_2, R_4, R_5$ ; $R_3$	2.57	1.88	1362	<b>78</b>	-	1	0	1	1
$R_3, R_4, R_5$ ; $R_2$	2.70	1.55	1746	<b>100</b>	-	0	1	1	1

Taula 5.4: Valor dels pins en funció del nivell de sensibilitat del sensor d'humitat

Per tal d'escollir 5 nivells de sensibilitat raonables es van fer proves amb el sensor, i es va decidir agafar-les una mica baixes —la sensibilitat de nivell 5 té una sensibilitat total de només el 33%— per evitar disparar el sensor amb el contacte involuntari amb la pell del pacient. Inicialment, es tenia pensat que el dispositiu electrònic pogués configurar la sensibilitat del sensor en una escala de 5 nivells, però les primeres proves pilot sobre el prototip obtingut van mostrar que no hi havia una diferència perceptible entre nivells consecutius i per tant es va prendre la decisió de passar de 5 nivells de sensibilitat a 2 únics nivells —sensibilitat baixa i sensibilitat alta— corresponents al nivell 1 i al nivell 5 respectivament.

Així doncs, establint el valor adequat als pins del microcontrolador es pot definir el nivell de sensibilitat del sensor d'humitat. D'això se n'encarrega la funció *void change\_sensitivity(char sen)* del programa *MAIN.C*, disponible en l'annex B.5.

### 5.3.4 Balanç energètic

Fer un balanç energètic del dispositiu electrònic permet estimar el seu temps de vida. Per a dur a terme l'estimació, s'han mesurat els diferents consums del dispositiu en funció de l'estat en que es troben el microcontrolador i el transmissor Bluetooth, ja que són els components que més consumeixen. Els valors obtinguts són els mostrats en la taula 5.4, on s'evidencia un elevat consum en el cas en que aquests estan treballant (*despert+ON*), en contraposició amb un consum pràcticament negligible quan aquests no estan fent cap

### 5.3. PREPARACIÓ DEL DISPOSITIU

tasca (*adormit+OFF*).

Estat dispositiu	Consum mesurat
Micro despert + Transmissor Bluetooth ON (sense transmetre)	20 mA
Micro despert + Transmissor Bluetooth ON (transmetent)	47 mA
Micro adormit + Transmissor Bluetooth OFF	18 $\mu$ A

Taula 5.5: Consum del dispositiu electrònic segons el seu estat

Els consums mesurats en l'estat *despert+ON* concorden amb els valors dels datasheets del fabricant del *PBA31308/2* [63] i del *XC864* [68]. Per a l'estat *adormit+OFF*, aquest valor difereix una mica del teòric ja que en tractar-se d'un consum tant baix també cal considerar el consum de la porta lògica i de la resta de components del dispositiu. Per tant, el consum d'aquests elements és aproximadament la diferència entre el consum mesurat (18 $\mu$ A) i la suma del consum teòric del transmissor Bluetooth en *OFF* (1 $\mu$ A) i el consum teòric del microcontrolador en mode *powerdown* (10 $\mu$ A), és a dir, uns 7 $\mu$ A.

Tenint en compte que segons el datasheet del fabricant [75] la bateria *CR1/3N* del dispositiu té una capacitat nominal de 160mAh, suposant una búsqueda inicial de terminals *PPStop* —de durada aproximada de 6 segons— i considerant una activació d'alarma diària en el pitjor dels casos —típic d'un pacient que rep aquest tipus de tractament [3]— i de durada de connexió de 3 segons, es pot fer la següent estimació:

$$bateria = 160mAh = 576000mAs \quad (5.4)$$

$$búsqueda de terminals = 20mA \cdot 3s + 47mA \cdot 3s = 201mAs \quad (5.5)$$

$$alarma diària = 20mA \cdot 2s + 47mA \cdot 1s = 87mAs \quad (5.6)$$

$$resta del dia = 18\mu A \cdot 86397s = 1555.146mAs \quad (5.7)$$

Tenint en compte els càlculs 5.4, 5.5, 5.6 i 5.7, s'obté un temps de vida del dispositiu en el pitjor cas de:

$$576000mAs = 201mAs + (87mAs + 1555.146mAs) \cdot N \quad (5.8)$$

$$N \approx 350 \text{ dies} \approx 1 \text{ any}$$

Així doncs, per a un us raonable del dispositiu es té un temps de vida d'1 any, temps que compleix el requisit del sistema que fa referència al temps de vida del dispositiu.

### 5.3.5 Prototip final

Una vegada programada la unitat de control del microcontrolador, es va procedir a emplaçar tots aquells components que encara no s'havien incorporat per qüestions pràctiques —la bateria per no gastar-la durant la fase de programació i testeig del microcontrolador i el sensor d'humitat per permetre la grabació del programa de la unitat de control en el microcontrolador—; es va substituir l'alimentació de la font per la de la bateria *CR1/3N* i es va procedir a soldar el sensor d'humitat. Amb això, el dispositiu quedava del tot complert, i ja només feia falta encapsular-lo i enresinar-lo ben hermèticament per aïllar la placa electrònica de qualsevol tipus d'humitat que pugués malmetre el dispositiu.



Figura 5.12: Primer prototip PPStop

El disseny de l'encapsulat es pot consultar en l'annex B.7, on s'hi mostra el seu modelatge en 3 dimensions. Una vegada encapsulat, es va afegir una cinta per a permetre que el dispositiu es pugués posicionar fixament a la part genital del pacient. El resultat del primer prototip obtingut és el que es mostra en la figura 5.12.

# Capítol 6

## Banc de proves

### 6.1 Test i proves pilot

Una vegada obtingut el primer prototip, s’han realitzat diversos test i proves pilot de laboratori a l’Hospital Sant Joan de Déu amb l’objectiu d’analitzar diferents paràmetres de funcionament del sistema i de detectar un possible comportament incorrecte del sistema. Per a portar-les a terme, es van preparar 5 unitats idèntiques del prototip 6.1 amb el codi *PPStop* —adreça *MAC* del dispositiu— de cadascun d’ells, i es van adquirir 5 terminals *NOKIA* model 6124[76] —model compatible amb *J2ME* i dotat de connectivitat bluetooth— als quals se’ls va instal·lar l’aplicació de control *PPStop*.

Els diferents paràmetres analitzats han estat els següents:

1. Proves de funcionament global del sistema per a 1 i 2 terminals associats.
2. Proves de cobertura Bluetooth.
3. Proves de temps de resposta d’alarma per a 1 i 2 terminals associats.
4. Proves de sensibilitat del sensor d’humitat.
5. Proves de coexistència amb altres dispositius *PPStop*.

Els resultats obtinguts per les proves pilot al laboratori de l’hospital s’han centrat en el primer punt, mentre que els test fets en desenvolupament s’han centrat en tots ells. Les proves de funcionament global s’han passat satisfactòriament tal i com s’esperava. Les proves de sensibilitat del sensor d’humitat han permès verificar el correcte funcionament de l’opció de modificació de sensibilitat del sensor d’humitat, i percebre la diferència entre els diferents nivells de sensibilitat establerts. Les proves de cobertura han donat resultats coherents dins dels esperats segons les especificacions de Bluetooth vistes en l’apartat 3.1.2 i segons les especificacions del transmissor Bluetooth *PBA31308/2* d’Infineon B.1, que especifiquen que el *PBA31302* és un dispositiu de classe 2 i, per tant, per a aquesta classe el rang de cobertura esperable és d’aproximadament 10m. El rang de cobertura real observat en les proves ha estat d’uns 10-15 metres en funció del nombre d’obstacles

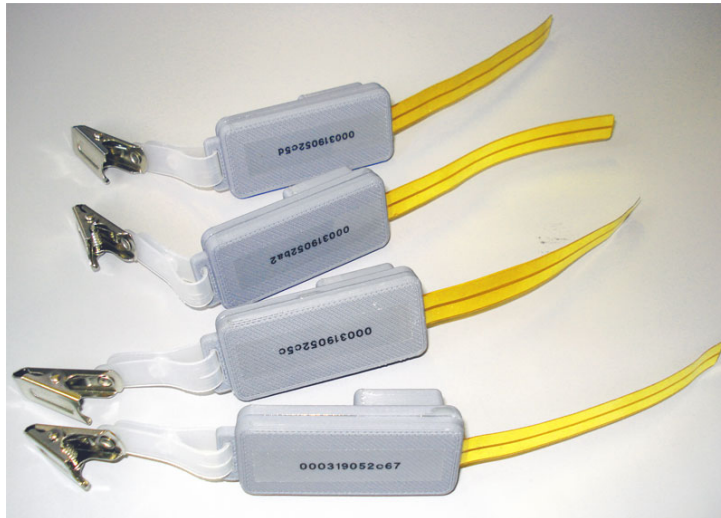


Figura 6.1: Pack de prototips per a les proves pilot

materials que es trobin enmig —persones, parets, etc— i de les interferències que puguin existir en la banda de freqüències on opera Bluetooth, i és suficient per a l'ús que se li ha de donar al sistema. Les proves de coexistència amb altres dispositius *PPStop* també han tingut èxit, ja que en el disseny del sistema es va contemplar aquesta possibilitat. Això s'analitza en més detall en l'apartat 6.3.

De totes elles, la prova més crítica ha estat la dels temps de resposta d'alarma per a 1 i 2 terminals, i més concretament per al primer terminal a alarmar ja que amb total certesa serà el terminal del pacient. És per això que els resultats obtinguts s'han analitzat en més detall en l'apartat 6.2, tot i que es pot concloure que en general els temps d'alarma compleixen els requisits de temps d'alarma citats en l'apartat 2.1.

## 6.2 Temps de resposta d'alarma

Per a fer les proves de temps de resposta d'alarma s'han emprat dos models diferents de terminal mòbil del fabricant *NOKIA*: el model 6124[76] i el model E50[77] també dotat de connectivitat Bluetooth però una mica més antic —de fet aquest model ja s'ha deixat de fabricar—. Amb això s'ha volgut aprofitar l'experiment per observar també si existeix alguna correlació entre el temps de resposta i el model de terminal.

L'experiment s'ha realitzat en un entorn domèstic, i s'han fet mesures per als següents casos:

1. Un únic terminal associat, model *NOKIA E50*, a 1m de distància en visió lliure.
2. Un únic terminal associat, model *NOKIA 6124*, a 1m de distància en visió lliure.
3. Un únic terminal associat, model *NOKIA E50*, a 5m de distància en visió lliure.



## 6.2. TEMPS DE RESPOSTA D'ALARMA

4. Un únic terminal associat, model *NOKIA 6124*, a 5m de distància en visió lliure.
5. Un únic terminal associat, model *NOKIA E50*, a 10m de distància sense visió lliure.
6. Un únic terminal associat, model *NOKIA 6124*, a 10m de distància sense visió lliure.
7. Dos terminals associats, models *NOKIA 6124*(1r) i *NOKIA E50*(2n), ambdós a 1m de distància en visió lliure.
8. Dos terminals associats, models *NOKIA 6124*(1r) i *NOKIA E50*(2n), el 1r a 1m de distància i el 2n a 5m de distància, ambdós a 1m de distància en visió lliure.
9. Dos terminals associats, models *NOKIA 6124*(1r) i *NOKIA E50*(2n), el 1r a 1m de distància en visió lliure i el 2n a 10m de distància sense visió lliure.
10. Dos terminals associats, models *NOKIA 6124*(1r) i *NOKIA E50*(2n), el 1r fora de cobertura i el 2n a 1m de distància en visió lliure.
11. Dos terminals associats, models *NOKIA 6124*(1r) i *NOKIA E50*(2n), el 1r fora de cobertura i el 2n a 5m de distància en visió lliure.
12. Dos terminals associats, models *NOKIA 6124*(1r) i *NOKIA E50*(2n), el 1r fora de cobertura i el 2n a 5m de distància sense visió lliure.

Com a instrument de mesura s'ha fet servir un cronòmetre convencional amb resolució de centèssimes de segon. Donat que en la presa de mesures pot existir un error de precisió en l'engegada i parada del cronòmetre —factor humà—, s'han agafat tres mostres diferents de cada mesura i s'han repetit aquelles mesures que per pròpia inspecció eren susceptibles de ser considerades errors aberrants. Un cop agafades les tres mostres de cada mesura, s'ha procedit al càlcul de la mitjana per a així fer-ne la representació gràfica. Els resultats obtinguts es mostren en la taules 6.1, 6.2 i les gràfiques de la figura 6.2, on "KO" indica el cas en que el terminal està fora de rang de cobertura i "OK" indica que el terminal està en cobertura i amb l'aplicació de control *PPStop* engegada.

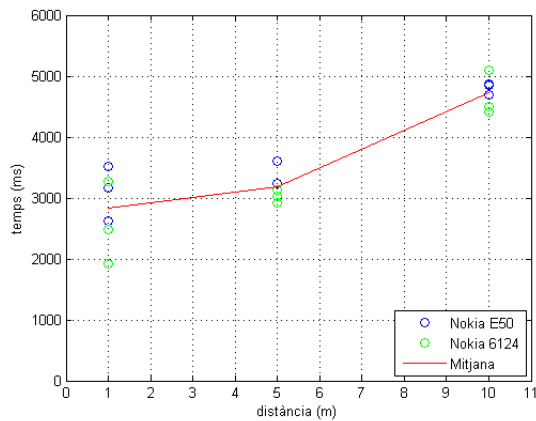
model \ dist.	1 terminal			2 terminals					
	1m	5m	10m	1r ok, 2n ko			1r ko, 2n ok		
				1m	5m	10m	1m	5m	10m
<b>Nokia E50</b>	3510	3130	4860	8860	8800	11190	9480	9370	11490
	3170	3610	4850	9120	8790	10940	9350	10010	12110
	2620	3240	4690	8130	9230	11350	8670	9660	11450
<b>Nokia 6124</b>	3280	3140	5100	3280	3140	5100	-	-	-
	1920	2920	4500	1920	2920	4500	-	-	-
	2490	3030	4410	2490	3030	4410	-	-	-

Taula 6.1: Mostres de temps(ms) d'alarma segons distància, per a 1 i 2 terminals

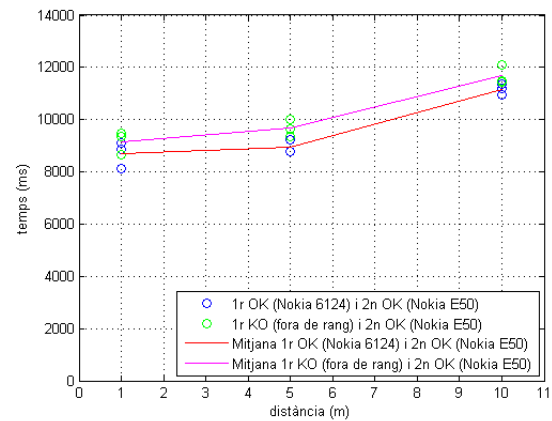
A partir d'aquests resultats es poden treure diverses conclusions interessants, tant de l'experiment amb un terminal associat com amb dos terminals associats.

model \ dist.	1 terminal			2 terminals					
	1m	5m	10m	1r ok, 2n ko			1r ko, 2n ok		
Nokia E50	3100	3330	4800	8700	8940	11160	9170	9680	11680
Nokia 6124	2560	3030	4670	2560	3030	4670	-	-	-

Taula 6.2: Mitjanes de temps(ms) d'alarma segons distància, per a 1 i 2 terminals



(a) Un únic terminal associat



(b) Dos terminals associats

Figura 6.2: Gràfica del temps d'alarma en funció de la distància

La primera de totes és que a priori sembla que el model de terminal pot arribar a incidir en el temps d'alarma. En el cas que s'ha experimentat, el model *NOKIA E50* triga gairebé mig segon més en generar l'alarma que el model *NOKIA 6124*. A priori no hi ha una explicació lògica per a aquest fet, però podria ser o bé perquè en ser un model més antic triga més en atendre les connexions Bluetooth, o bé perquè en general té una capacitat de procés inferior. En qualsevol cas, s'ha considerat que la introducció d'un retard inferior als 0.5 segons no és rellevant i no s'ha donat més importància a aquest fet. En segon lloc, s'ha pogut observar com la presència d'obstacles materials enmig de la comunicació bluetooth produeix —tal i com es preveia— un augment del temps d'alarma. Aquest fet s'evidencia amb la no-linealitat de les gràfiques de la figura 6.2. També d'aquestes gràfiques se n'extreu fàcilment que existeix una dependència aproximadament lineal entre la distància i el temps d'alarma, per a un mateix entorn —en aquest cas un entorn de visió lliure—. La última, i probablement més important, és que el temps de resposta d'alarma per a distàncies inferiors a 1m —cas real d'un pacient d'enuresi— està generalment entorn els 3 segons, que és precisament el requisit que es demanava al sistema.

De l'experiment amb més d'un terminal se'n poden extreure conclusions referents al temps de *timeout* del transmissor bluetooth quan aquest no pot establir una connexió. Per entendre-ho millor, és important tenir present la gràfica 3.11 —corresponent al cas

## 6.2. TEMPS DE RESPOSTA D'ALARMA

---

d'alarma a 2 terminals *KO/OK*— mostrada en el capítol 3, que il·lustra exactament quines accions es consideren en el temps d'alarma. A partir de la taula de mitjanes 6.2 s'han pogut fer els càlculs que es mostren a continuació.

Havent obtingut el temps d'alarma per al model *NOKIA E50* a 1m de distància en el cas d'un sol terminal associat ( $t_1$ ), i el temps d'alarma per al 2n terminal a 1m de distància en el cas *KO/OK* quan es tenen dos terminals associats ( $t_2$ )

$$t_1 = t_{despertat} + t_{configuracio} + t_{connexio} = 3100ms \quad (6.1)$$

$$t_2 = t_{despertat} + t_{configuracio} + t_{timeout} + t_{connexio} = 9170ms \quad (6.2)$$

s'ha pogut establir la següent equivalència i obtenir una estimació del temps de *timeout* ( $t_{timeout}$ ):

$$t_2 = t_{timeout} + t_1 \quad (6.3)$$

$$t_{timeout} = 9170 - 3100 \approx 6s \quad (6.4)$$

Per obtenir una estimació del  $t_{despertat} + t_{configuracio}$  ha calgut recórrer a resultats obtinguts amb el simulador PPStop vist en l'apartat 5.2.4 del capítol 5, ja que el simulador utilitza el transmissor Bluetooth *PBA31308/2* enviant-li directament les comandes AT. En el simulador es va incorporar un comptador de temps per determinar el temps total de connexió, és a dir, des que s'inicia la connexió fins que es fa la desconnexió. Els resultats obtinguts eren mostrats en el *log* del simulador, quedant enregistrats en les traces generades per aquest. Tenint en compte això i tenint en compte les traces de l'apartat (e) de la taula 5.3 —cas de generació de 2 alarmes consecutives a un únic terminal associat, model *NOKIA E50*—, s'ha pogut estimar un temps de connexió ( $t_{connexio}$ ) per al terminal *NOKIA E50* quan aquest es troba a 1m de distància del transmissor Bluetooth.

$$t_{connexio} = \sum_{i=1}^2 \frac{t_{connexio_i}}{2} = \frac{968ms + 1578ms}{2} \approx 1300ms \quad (6.5)$$

Amb aquest valor, i fent ús de les expressions 6.1 o 6.2 es pot estimar finalment el  $t_{despertat} + t_{configuracio}$ :

$$t_{despertat} + t_{configuracio} = t_1 - t_{connexio} \approx 3100ms - 1300ms \approx 1.8s \quad (6.6)$$

En resum, és té un temps de timeout d'uns 6 segons i un temps de despertat de gairebé 2 segons ja que el temps de configuracio se sap que és negligible per tractar-se de simples comandes AT de configuració. El primer resultat era esperable segons les especificacions del *PBA31308/2*[63], en canvi per al segon no es pot valorar ja que les especificacions no en diuen res, però és un temps a tenir molt en compte i que no es pot eliminar ja que per minimitzar el consum d'energia del dispositiu electrònic el procés de *power-off* i, consegüentment, el de *power-on* són inevitables.

En conclusió, els temps d'alarma per al primer terminal generalment compleix els requisits però el seu valor depen més del temps de despertat que del propi temps de connexió Bluetooth. En els temps d'alarma per a un segon terminal, a més a més cal tenir en compte el factor del temps de *timeout* en el cas que el primer terminal no estigui en cobertura, però cal remarcar que aquest cas és difícil que es doni ja que per norma general el dispositiu electrònic i el primer terminal —terminal del pacient— sempre estaran a menys d'1m de distància.

### 6.3 Coexistència amb altres dispositius

Per a fer les proves de coexistència amb altres dispositius s'han emprat dos dispositius electrònics i dos terminals mòbils —mínims elements necessaris per a provar tots els casos possibles— i s'ha comprovat que el funcionament del sistema era l'esperat. S'entén per coexistència amb altres dispositius, la coexistència d'un dispositiu electrònic amb un altre dispositiu electrònic dins d'un mateix rang de cobertura, i la coexistència de més d'un terminal mòbil *PPStop* amb diferents configuració —és a dir, associats cadascun a un dispositiu electrònic diferent— dins d'un mateix rang de cobertura. Aquestes possibles situacions reals en que es pot trobar el sistema s'il·lustren clarament en la figura 6.3.

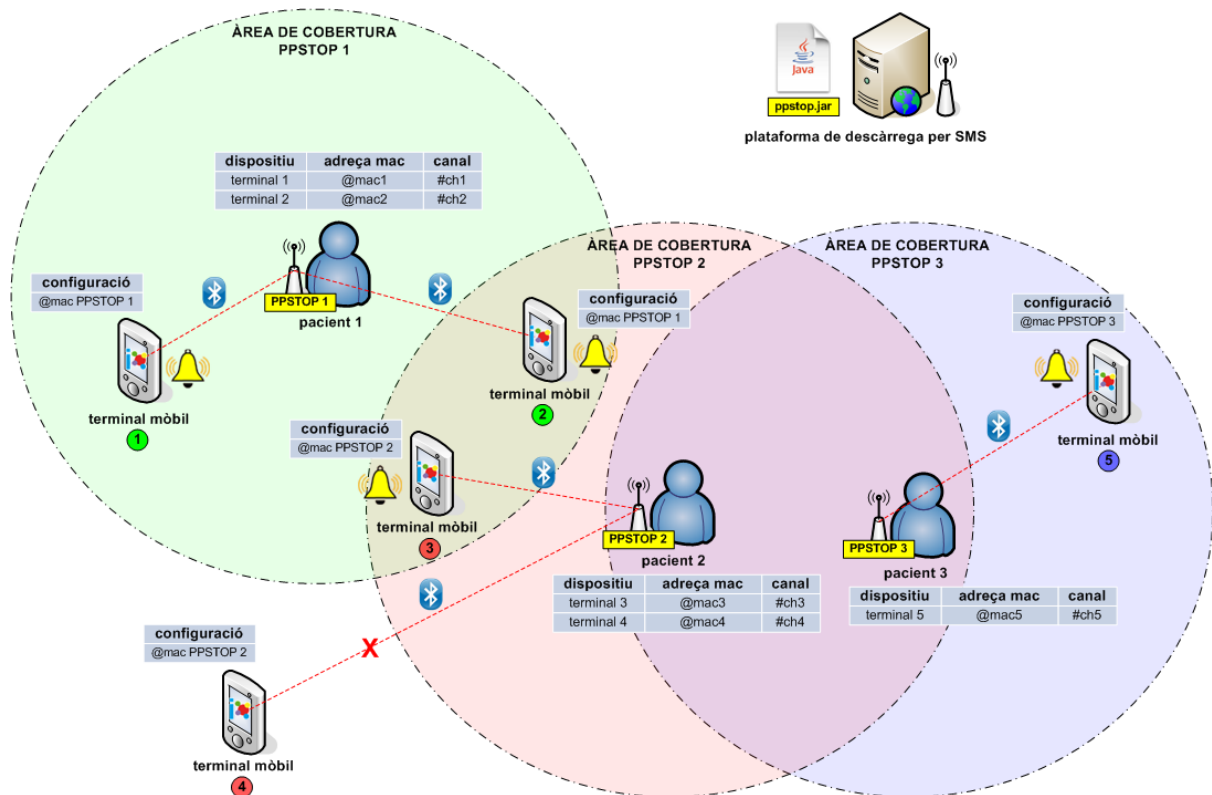


Figura 6.3: Escenari global real amb possible coexistència amb altres dispositius PPStop

### 6.3. COEXISTÈNCIA AMB ALTRES DISPOSITIUS

---

Els experiments realitzats han estat els següents:

1. Un dispositiu electrònic i dos terminals mòbils *PPStop* dins d'un mateix rang de cobertura, únicament un d'ells configurat amb el codi —adreça *MAC*— del dispositiu electrònic esmentat.
2. Dos dispositius electrònics i dos terminals mòbils *PPStop* dins d'un mateix rang de cobertura, un d'ells configurat amb el codi d'un dels dispositius electrònics esmentats i l'altre configurat amb el codi de l'altre dispositiu.

Els resultats obtinguts han sigut satisfactoris. La coexistència de dispositius electrònics i terminals mòbils *PPStop* associats a diferents dispositius electrònics no ha suposat cap problema en el punt del protocol que es tracta aquesta qüestió, que és la posterior verificació dels serveis trobats pel protocol *SDP* en el procés de búsqueda de terminals *PPStop* per tal de generar en memòria del dispositiu electrònic la llista de terminals *PPStop* associats.



# Capítol 7

## Aplicació de seguiment

### 7.1 Java 2 Standard Edition (J2SE)

Com ja s'ha comentat en el capítol *sec:cap4* La plataforma *JAVA* de *Sun Microsystems* permet el desenvolupament i l'execució de programes escrits en el llenguatge de programació *JAVA*. La plataforma no és ni específica d'un processador ni d'un sistema operatiu, si no més aviat un motor d'execució anomenat màquina virtual i un compilador amb una sèrie de llibreries estàndar que estan implementades per a hardware divers i diferents sistemes operatius, la qual cosa fa que els programes *JAVA* puguin executar-se en qualsevol plataforma.

Concretament la versió *J2SE* de *JAVA*, en comparació a altres versions disponibles com ara *J2ME* o *J2EE*, està orientada al desenvolupament d'aplicacions que han de ser executades en estacions de treball, és a dir, màquines que no són de recursos limitats.

#### 7.1.1 Introducció a J2SE

La *Java Platform Standard Edition*, més coneguda com a *J2SE* fins a la seva versió 5.0, és la plataforma utilitzada per a desenvolupar aplicacions *JAVA* de propòsit general. En termes pràctics i tal com il·lustra la figura 7.1, *J2SE* consisteix en una màquina virtual que és utilitzada per a executar programes *JAVA*, juntament amb una col·lecció de llibreries —també conegudes amb el nom de *packages* o APIs— necessàries per permetre l'ús de certs recursos com ara sistemes de fitxers, xarxes, interfícies gràfiques, etc. des dels programes que executa.

Començant amb la versió *J2SE 1.4* —*Merlin*—, la plataforma *JAVA SE* ha estat desenvolupada sota la supervisió del *Java Community Process*. L'especificació per a *JSE 1.4* és la *JSR-59* i la *JSR-176* especifica la versió 5.0 de *J2SE* —*Tiger*—. Des de 2006, *JAVA SE 6* —*Mustang*— està sent desenvolupada sota el *HSR-270*.

*J2SE* proporciona un gran conjunt de paquets amb llibreries molt extenses i cadascuna

d'elles amb una finalitat diferent. Les principals, i més utilitzades, són les que es mostren en la taula 7.1. Moltes d'elles s'han utilitzant per al desenvolupament de l'aplicació de seguiment de *PPStop*, però també ha calgut la inclusió de llibreries opcionals per a propòsits més concrets. Aquestes es detallen en la secció 8.2.

Paquet	Descripció del paquet
<i>java.lang</i>	Classes fonamentals i interfícies fortament relacionades amb el llenguatge i el sistema runtime.
<i>java.io</i>	Classes per al suport d'operacions d'entrada/sortida.
<i>java.nio</i>	(Nova IO) Classes per a suport d'entrada/sortida mapejada a memòria.
<i>java.math</i>	Classes per al suport d'aritmètica de multiprecisió.
<i>java.net</i>	Classes que proporcionen mètodes per a rutines especials d'entrada/sortida per a xarxes.
<i>java.text</i>	Classes que implementen rutines de tractament de cadenes de caràcters i altres.
<i>java.util</i>	Estructures de dades que afegixen objectes.
<i>java.applet</i>	Classes per al suport de creació d' <i>applet</i> Java.
<i>java.beans</i>	Classes per desenvolupar i manipular <i>beans</i> de <i>JavaBeans</i> .
<i>java.awt</i>	(Abstract Window Toolkit) Classes per al suport d'operacions bàsiques GUI.
<i>java.rmi</i>	Classes per a suport a la invocació de mètodes remots.
<i>java.security</i>	Classes per a operacions relacionades amb la seguretat.
<i>java.sql</i>	Es tracta d'una implementació de la API JDBC per a accés a bases de dades.
<i>javax.rmi</i>	Classes per al suport a la comunicació remota entre aplicacions.
<i>javax.swing</i>	Classes per al suport de creació de GUIs sobre <i>AWT</i> independents de la plataforma d'execució.

Taula 7.1: Paquets de Java Platform Standard Edition

### 7.1.2 JavaDB/Derby technology database

Quan es desenvolupa una aplicació sovint es necessita emmagatzemar dades referents a l'aplicació per tal de poder-les recuperar en qualsevol moment, no només durant l'execució de l'aplicació si no també en execucions posteriors. La persistència de dades en aplicacions *JAVA* es pot aconseguir emmagatzemant les dades en una base de dades —sigui remota o local— a la qual s'hi accedeix des de l'aplicació en el moment que es necessiti consultar, inserir, actualitzar o esborrar dades. No obstant, això planteja el problema de la dependència de l'aplicació amb la base de dades per al seu correcte funcionament. És habitual en els programadors d'aplicacions la necessitat de tenir un sistema gestor de base de dades que no sigui dependent de la plataforma on s'executi el programa i que



## 7.1. JAVA 2 STANDARD EDITION (J2SE)

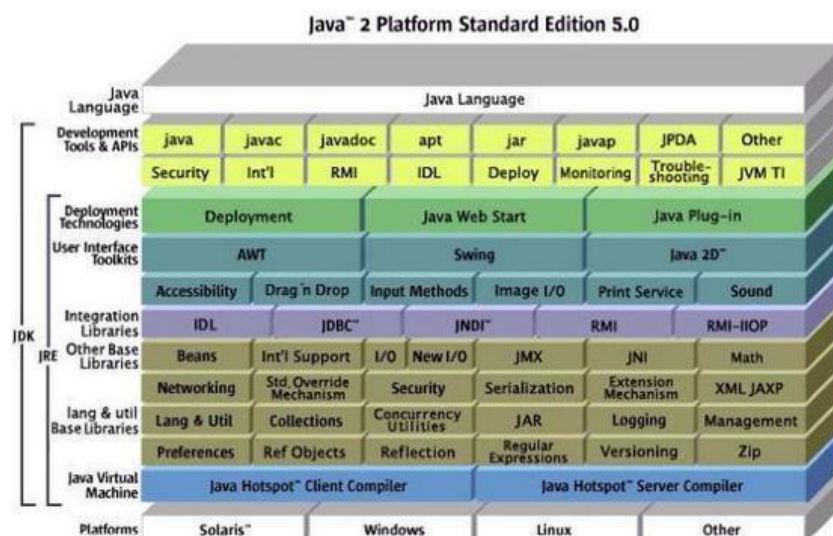


Figura 7.1: Java 2 Platform Standard Edition 5

sigui accessible en tot moment des de l'aplicació, és a dir, que sigui local; encastada a l'aplicació.

Per a resoldre aquesta problemàtica en la persistència de dades en aplicacions *JAVA*, Sun Microsystems va anunciar l'aparició de *JavaDB*[19], basat 100% en tecnologia *JAVA* que permetia la inclusió del sistema gestor de base de dades de codi obert *Apache Derby*[78] incrustada en les aplicacions *JAVA*. *JavaDB* és lleuger i és incrustable en una aplicació *JAVA*, permetent que aquesta pugui accedir-hi per a abastar funcionalitats d'emmagatzematge, creació de disparadors, creació de procediments i suport per al llenguatge *SQL*, software Java DataBase Connectivity (*JDBC*) i suport per a les plataformes *J2SE* i *J2EE*; tot això incrustat a la mateixa màquina virtual *JVM*. *JavaDB* únicament necessita la càrrega d'un driver *JDBC*[79] per a *Derby* a l'aplicació des de la qual es vol tenir accés a la base de dades —veure figura 7.2— i que permet l'accés i execució d'operacions sobre la base de dades.

Les característiques principals de *JavaDB* són:

- APIs per a *JDBC* i *SQL*. Suporta totes les característiques de *SQL92* i la majoria de *SQL99* —la sintaxis *SQL* utilitzada rové de *IBM DB2*—
- El seu codi comprimit ocupa vora els 2MB.
- Suporta xifratge complet, rols i permisos. A més a més poseeix esquees *SQL* per a separar la informació en una única base de dades i control complet d'usuaris.
- Internament suporta procediments, xifratge i compressió.
- Porta incorporat suport multilinguatge i localitzacions específiques.
- A partir de la versió 10.4 incorpora un sistema simple de rèpliques mestre-esclau.
- Transaccions i recuperació davant d'errors *ACID* —*Atomicity, Consistency, Iso-*

*lation, Durability—*

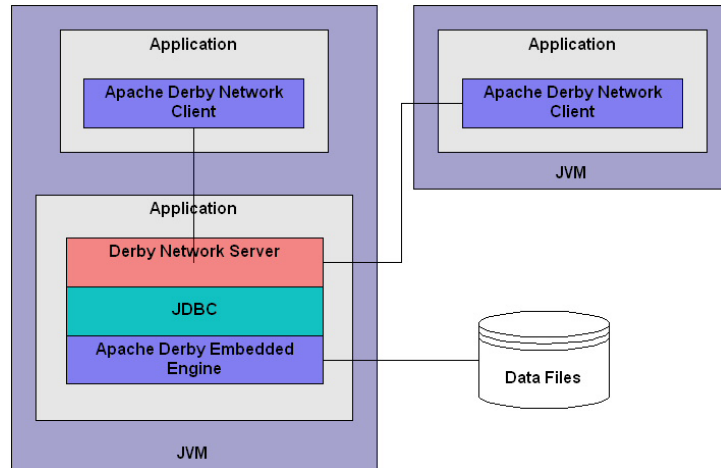


Figura 7.2: Utilització de JavaDB per a persistència de dades

Així doncs, en el cas de l'aplicació de seguiment *PPStop* ha calgut fer ús d'aquesta tecnologia ja que es necessitava persistència de les dades dels pacients i dels seus historials d'alarmes.

## 7.2 Paquets opcionals per a J2SE

Tal i com s'ha comentat, a banda dels paquets comuns a la edició *J2SE* n'existeixen altres d'opcionals que afegeixen funcionalitats extres. A continuació s'analitzen els paquets extres que s'han utilitzat en el projecte per al desenvolupament de l'aplicació de seguiment.

### 7.2.1 BlueCove API

*BlueCove*[17] és una implementació de l'estàndar *JSR-82* per a *J2SE* i per tant subministra una *API* Java per a Bluetooth *JSR-82* que permet utilitzar aquesta tecnologia des de la plataforma *JAVA* ja que la versió estàndar d'aquesta no posseeix llibreries pròpies per a Bluetooth. Aquesta *API* actualment actua correctament amb *Mac OS X*, *WIDCOMM*, *BlueSoleil* i la pila Bluetooth de Microsoft que es troba en el sistema operatiu *Windows versió XP SP2* i superiors. Originalment va ser desenvolupada per investigadors d'*Intel* però actualment està mantinguda per voluntaris.

Aquesta llibreria s'ha inclòs a l'aplicació de seguiment per a permetre realitzar una connexió Bluetooth a una determinada adreça —especificada pel codi d'historial del pacient, que equival a l'adreça física Bluetooth del seu terminal mòbil— i així poder dur a terme l'obtenció de l'historial d'alarmes del pacient.

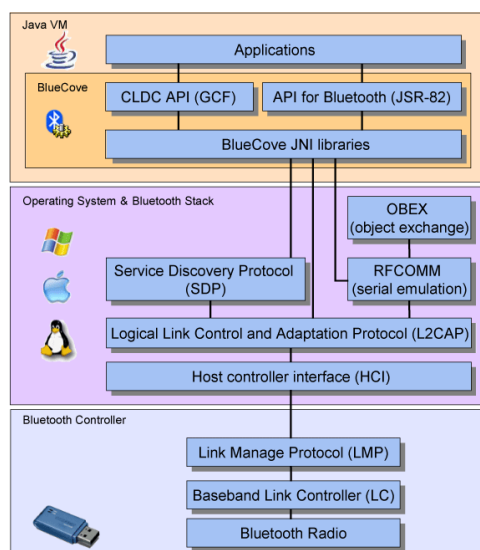


Figura 7.3: Llibreria BlueCove per a Bluetooth

### 7.2.2 JFreeChart/JCommon API

*JFreeChart*[20] és una de les llibreries més utilitzades per a la generació de gràfiques des de la plataforma *JAVA*, per facilitar molt la tasca del programador a l'hora de generar gràfiques de qualitat professional per a les seves aplicacions i per posseir una molt bona documentació[80] plena d'exemples d'utilització i personalització [81]. Ha estat dissenyada per David Gilbert per a ser utilitzada en el desenvolupament d'aplicacions *JAVA*, servlets i *JSP*, i es distribueix juntament amb el seu codi font per tractar-se d'una llibreria subjecta a *GNU-LGPL* la qual cosa permet que pugui ser utilitzada gratuïtament tant en aplicacions d'àmbit personal com en les d'àmbit comercial.

Un del seu major potencial és l'elevat nombre de tipus de gràfics que permet crear, tant en 2D com en 3D. Amb *JFreeChart* es poden crear gràfiques de barres, gràfiques circulars —comunment coneguts com a gràfics d'estil formatge—, gràfics lineals, diagrames de dispersió, gràfiques de series de temps, diagrames de Gantt, gràfiques polars, etc. Per al seu correcte funcionament únicament requereix de la versió 1.3.1 o posterior de la Java Development Kit (JDK), i la inclusió d'una segona llibreria anomenada *JCommon*[82] comuna a totes les aplicacions desenvolupades per *JFree* —el grup de desenvolupadors de *JFree*— i que aporta classes miscel·lànies per a donar suport a:

- Codi d'administració de configuració i dependències.
- Un *framework* general de registres.
- Utilitats de text.
- Classes d'interfície d'usuari per a mostrar informació sobre aplicacions.
- Personalització d'administradors de capes.
- Panell de selecció de dates.

- Utilitats per a serialització.

Aquestes dues llibreries s'han inclòs en l'aplicació de seguiment *PPStop* per a cobrir la necessitat de realitzar gràfiques en temps d'execució, concretament les gràfiques estadístiques dels historials dels pacients que es mostren en la pantalla principal de l'aplicació i en l'informe de pacient en PDF.

### 7.2.3 iText PDF API

*iText PDF* és una llibreria distribuïda sota llicència *MPL-LGPL* i que ha estat escrita escrita per Bruno Lowagie, Paulo Soares i altres. *iText PDF* permet generar i manipular fitxers PDF "al vol"—és a dir en temps d'execució— des d'aplicacions *JAVA*. Els fitxers *PDF* tenen la virtut de ser documents que poden incorporar text, llistes, taules i imatges i no depenen de cap plataforma, és a dir, poden ser visualitzats des de qualsevol sistema operatiu que poseeixi un entorn gràfic.

La biblioteca, a més a més, dona capacitats al programador per a omplir formularis dinàmicament, moure pàgines d'un *PDF* a un altre, dona suport a firmes basades en Public Key Infrastructure (PKI) de *PDF*, xifratge de 40 bits i 128 bits, correcció de colors, *PDF/X*, gestió de colors per perfils *ICC* i altres.

Per al seu correcte funcionament la llibreria requereix la versió 1.5 o superior de la *JDK*, i en el cas de l'aplicació de seguiment *PPStop* s'ha hagut de fer servir per cobrir la necessitat de generar els informes dels pacients en format PDF, cosa que permet obtenir i emmagatzemar, imprimir o distribuir el seguiment del tractament del pacient de manera visualment rica i pràctica.

## 7.3 Desenvolupament de l'aplicació

En aquest apartat es descriu l'estructura que s'ha seguit en el desenvolupament de l'aplicació del PC —aplicació de seguiment— amb el llenguatge *J2SE* les característiques i llibreries del qual s'han analitzat anteriorment.

### 7.3.1 Entorn de desenvolupament

De la mateixa manera que per a la programació de l'aplicació *J2ME* del terminal mòbil, per a la programació amb el llenguatge *J2SE* s'ha fet servir l'entorn de programació *NetBeans*[58] —versió 6.7—. Existeixen altres tipus d'entorns de programació, però el principal avantatge que té *Netbeans* és que està disponible tant per a la plataforma *Windows* com *Linux* i proporciona eines i mòduls excel·lents que faciliten molt la vida del programador, i més concretament en el desenvolupament d'aplicacions *JAVA*.

## 7.3. DESENVOLUPAMENT DE L'APLICACIÓ

El paquet J2SDK instal·la una màquina virtual *JAVA* i les llibreries de *J2SE* necessàries per al desenvolupament d'aplicacions amb aquest llenguatge. A més a més, també s'han afegit les llibreries *BlueCove*, *JFreeChart* i *iText PDF* analitzades en els apartats 7.2.1, 7.2.2 i 7.2.3 respectivament i que ha calgut utilitzar donades les necessitats de l'aplicació.

### 7.3.2 Disseny de la Base de Dades de JavaDB

Prèviament al desenvolupament de la lògica del programa, s'ha procedit al disseny de la base de dades de *JavaDB* en funció de les necessitats de l'aplicació. Les necessitats de persistència de dades per a l'aplicació són les següents:

1. Emmagatzemar l'identificador de l'idioma de l'aplicació —català, castellà o anglès— seleccionat per l'usuari.
2. Emmagatzemar les dades de tots els pacients que es donin d'alta a l'aplicació.
3. Emmagatzemar les alarmes de cada pacient, és a dir, l'historial d'alarmes de cada pacient.

A partir d'aquí, s'ha decidit utilitzar una taula per a cada tipus d'emmagatzematge; una per a emmagatzemar els paràmetres de configuració —en aquest cas únicament l'idioma—, una altra per emmagatzemar a tots els pacients i una última per a emmagatzemar totes les alarmes. A més a més, seguint el model relacional i sabent que a un pacient li poden correspondre diverses alarmes, s'ha establert una relació binària de 1 a N entre la taula de pacients i la taula d'alarmes, tal i com mostra la figura 7.4. Com es pot comprovar, els noms assignats a les tres taules han estat *Config*, *Pacients* i *Alarmes*, respectivament.

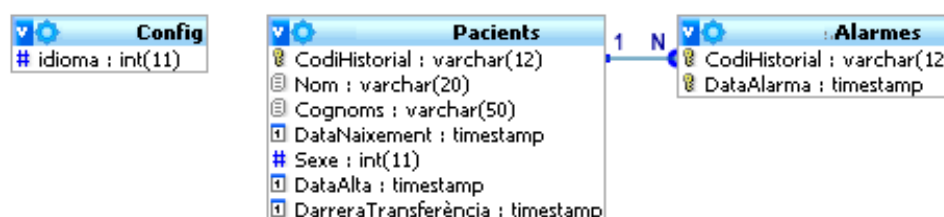


Figura 7.4: Disseny de la base de dades de l'aplicació

Les taules 7.2 i 7.3 mostra les restriccions i la descripció dels atributs de les taules *Pacients* i *Alarmes*. Com a clau primària de la taula *Pacients* s'ha escollit l'atribut (*CodiHistorial*), ja que cada dispositiu *PPStop* només podrà ser utilitzat per un únic pacient i per tant defineix un identificador únic i de valor no nul dins de la taula. Com a clau primària de la taula *Alarmes* s'han escollit els atributs (*CodiHistorial*, *DataAlarma*) ja que defineixen un identificador únic i no nul en no poder-se produir dos alarmes en el mateix instant per a un sol pacient. A més a més, l'atribut *CodiHistorial* és clau forana i referencia a l'atribut *CodiHistorial* de la taula pacients ja que la taula només

emmagatzemarà alarmes de pacients que estiguin registrats a l'aplicació, i per tant, a la base de dades.

Aquesta darrera clau forana a més ha de tenir la propietat *ON CASCADE DELETE* ja que volem que en eliminar-se un pacient de la taula *Pacients* automàticament s'eliminin les seves alarmes emmagatzemades a la taula *Alarmes*.

Nom del camp	Restricció	Descripció del paquet
<i>CodiHistorial</i>	Clau primària	Codi d'historial del pacient
<i>Nom</i>	—	Nom del pacient
<i>Cognoms</i>	—	Cognoms del pacient
<i>DataNaixement</i>	—	Data de naixement del pacient
<i>Sexe</i>	—	Identificador del sexe del pacient
<i>DataAlta</i>	—	Data d'alta del pacient
<i>DarreraTransferència</i>	—	Data de la darrera transferència d'historial

Taula 7.2: Restriccions i descripció dels atributs de la taula "Pacients"

Nom del camp	Restricció	Descripció del paquet
<i>CodiHistorial</i>	Clau primària, Clau forana	Codi d'historial del pacient
<i>DataAlarma</i>	Clau primària	Data de l'alarma

Taula 7.3: Restriccions i descripció dels atributs de la taula "Alarmes"

Donades les característiques del sistema gestor de base de dades *Apache Derby* i el llenguatge *SQL*, el codi que utilitza l'aplicació de seguiment per a crear les taules a la base de dades és el mostrada a 7.3.2:

Codi 7.1: Codi SQL de creació de taules per a JavaDB

```

1 CREATE TABLE "Pacients"(
2     "CodiHistorial" VARCHAR(12),
3     "Nom" VARCHAR(20),
4     "Cognoms" VARCHAR(50),
5     "DataNaixement" TIMESTAMP,
6     "Sexe" INTEGER,
7     "DataAlta" TIMESTAMP,
8     "DarreraTransferència" DATE,
9     PRIMARY KEY ("CodiHistorial")
10 );
11
12 CREATE TABLE "Alarmes"(
13     "CodiHistorial" VARCHAR(12),
14     "DataAlarma" TIMESTAMP,
15     PRIMARY KEY ("DataAlarma","CodiHistorial"),
16     FOREIGN KEY ("CodiHistorial") REFERENCES "Pacients" ON DELETE CASCADE
17 );
18
19 CREATE TABLE "Config"(
20     "Idioma" INTEGER
21 );

```

### 7.3.3 Implementació del patró MVC

En el cas de l'aplicació *PPStop* de seguiment del tractament i igual com s'ha fet per a l'aplicació *J2ME* del terminal mòbil, s'ha seguit el patró de programació *MVC*. Aquest patró ja s'ha detallat en l'apartat 4.3.2; no obstant la implementació del patró en *J2SE* és més fàcil ja que a diferència de la versió microedition, aquesta té implementades la classe *java.util.Observable* i la interfície *java.util.Observer*.

Així, doncs, les classes que creen la interfície d'usuari —*GUI*— de l'aplicació de seguiment han d'implementar la interfície *Observer* i el model de dades ha d'extendre la classe *Observable*. Aquesta darrera classe té una sèrie de mètodes interessants:

**addObserver(Observer o)** Mètode de la classe *Observable* que afegeix un observador a la seva llista d'observadors, per a que aquest pugui ser notificat quan hi hagi un canvi en el model.

**setChanged():** Mètode de la classe *Observable* que invoca el model per a indicar que alguna de les dades definides en el model ha canviat.

**notifyObservers():** Mètode de la classe *Observable* que invoca el model per a indicar als seus observadors —la vista— que el seu estat ha canviat i que cal refrescar la vista. La crida d'aquest mètode comporta l'execució del mètode *update(Observable obs, Object o)* present en els observadors —el tindran implementat forçosament ja que és un mètode definit per la interfície *Observer*— que s'encarrega d'actualitzar els paràmetres de la vista amb els nous valors adquirits a través dels mètodes públics del model.

Amb la correcta crida d'aquests mètodes s'aconsegueix que l'aplicació segueixi el patró de manera satisfactòria.

### 7.3.4 Estructura i aparença

La principal funció de l'aplicació de seguiment *PPStop* és la de rebre historials d'alarmes de l'aplicació del terminal mòbil a través de Bluetooth per a posteriorment emmagatzemar-les i així poder mostrar tot l'historial d'alarmes d'un pacient i extreure estadístiques gràfiques que permetin a l'usuari dur a terme un correcte seguiment del tractament d'enuresi. Per tant, doncs, l'aplicació actua fonamentalment com a gestor de pacients i adquirent, arxivador i llistador de dades —les alarmes dels pacients—. Addicionalment l'aplicació també ha de ser capaç de generar informes en format PDF mostrant totes aquestes dades.

Havent recordat totes aquestes funcionalitats que ha de tenir l'aplicació, a aquest apartat es centra únicament en l'anàlisi de l'estructura de l'aplicació i el seu funcionament intern. Per a veure el funcionament de l'aplicació a nivell d'usuari s'ha elaborat una guia de l'usuari que es pot consultar a l'annex D.4.

### Estructura de paquets i classes

Internament, l'aplicació s'ha dividit en diferents paquets de classes, cadascun d'ells amb l'objectiu de cobrir unes determinades necessitats. Així ho mostra la taula 7.4.

Paquet	Classes	Funcionalitats que cobreix
<i>Bluetooth</i>	Bluetooth, Connexió	Conté les classes per a realitzar una connexió Bluetooth i gestionar tota comunicació amb l'aplicació
<i>Constants</i>	CONST, LANG	Conté les classes de constants numèriques i de text
<i>Dades</i>	Alarma, DB, Historial, Model, Pacient	Conté les classes per a gestionar i emmagatzemar totes les dades de l'aplicació
<i>Main</i>	PPStop	Conté la classe principal de l'aplicació
<i>Utils</i>	Utils, UtilsVista	Conté les classes amb funcions auxiliars de certa utilitat
<i>Vista</i>	DialogPacient, GUI, Splash-Screen, Vista	Conté les classes per a generar i gestionar la GUI (Graphic User Interface) de l'aplicació

Taula 7.4: Paquets de classes de l'aplicació de seguiment

Cada paquet, com es pot veure, conté una sèrie de classes i/o interfícies, i aquestes s'encarreguen d'una funció específica dins del paquet i, en general, dins de l'aplicació. A continuació es descriu la funcionalitat de cada classe dins de l'aplicació.

- **Bluetooth.Bluetooth:** Classe que obté l'historial d'alarmes d'un pacient *PPStop* mitjançant Bluetooth.
- **Bluetooth.Connexió:** Classe que obre els *streams* d'e I/O per a la transmissió i recepció de dades.
- **Constants.CONST:** Interfície que conté totes les constants numèriques utilitzades per l'aplicació.
- **Constants.LANG:** Interfície que conté totes les constants d'idioma utilitzades per l'aplicació.
- **Dades.Alarma:** Classe representativa d'una alarma
- **Dades.BD:** Classe que permet interactuar amb la base de dades JavaDB(derby).
- **Dades.Historial:** Classe representativa de l'Historial d'Alarmes.
- **Dades.Model:** Classe que conté totes les dades de l'aplicació.
- **Dades.Pacient:** Classe representativa d'un pacient.
- **Main.App:** Classe principal de l'aplicació de seguiment *PPStop*.



### 7.3. DESENVOLUPAMENT DE L'APLICACIÓ

- **Utils.Utils:** Classe amb funcions útils d'àmbit general.
- **Utils.UtilsVista:** Classe amb funcions útils en l'àmbit de la creació de la *GUI* (Graphic User Interface).
- **Vista.DialogPacient:** Classe que conté tota la GUI per a donar d'alta un nou pacient o modificar-ne un de ja existent.
- **Vista.GUI:** Classe que conté tota la gestió de la GUI (Graphic User Interface). Actua com a Vista dins del patró MVC (Model-View-Controller).
- **Vista.SplashScreen:** Classe que defineix la finestra Splash d'inici de l'aplicació.
- **Vista.Vista** Interfície d'una vista per a aplicar el patró MVC (Model-View-Controller).

Cada classe té uns determinats mètodes i atributs, que es poden consultar en el *JavaDoc* de l'aplicació que s'adjunta en l'annex D.2. Tanmateix, també es pot consultar el codi *JAVA* de l'aplicació —veure annex D.3— on es mostra com s'ha programat cada classe. Una vegada descrites les funcionalitats de les classes i interfícies, es passa a descriure breument quin és el seu funcionament. Per similitud de les estructures internes de les aplicacions del terminal mòbil i la de seguiment —veure figures 4.14 i 7.5—, no es tornarà a analitzar com és la interacció entre les classes tot seguint el patró *MVC* ja que ja s'ha comentat en la secció 4.3.3 del capítol 4.

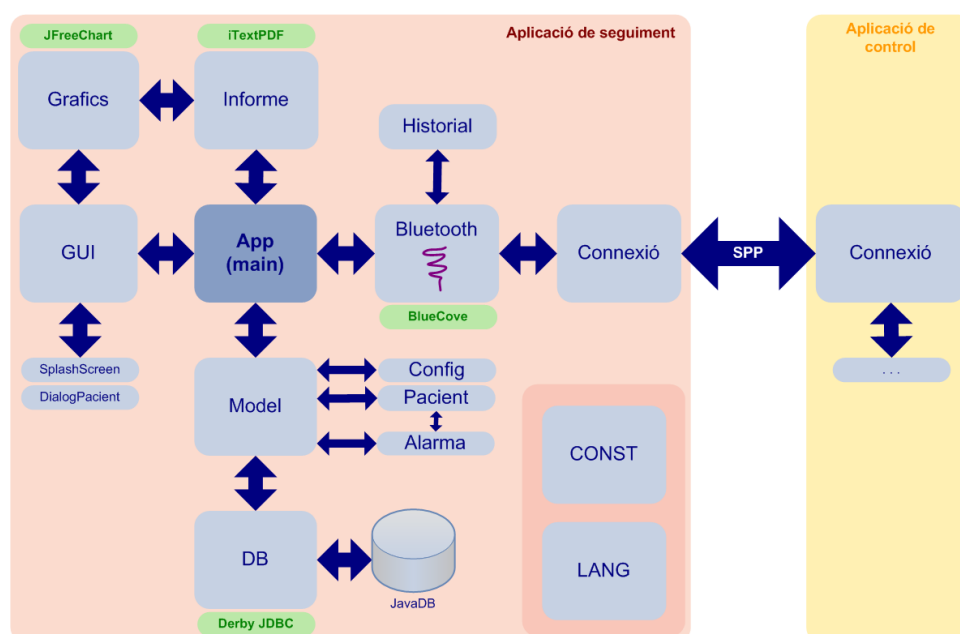


Figura 7.5: Estructura simplificada de l'aplicació de seguiment

## Aparença gràfica

L'entorn gràfic de l'aplicació s'ha desenvolupat a partir dels paquets *SWING* i *AWT* que incorpora *J2SE*, i que per si soles han permès un elevat nivell de personalització de l'interfície de l'usuari, adequada als requisits que es tenien per a l'aplicació. El resultat final és el que es mostra a la figura 7.6, on es pot apreciar el nivell d'informació que de manera visual es dona a l'usuari final i també la simplicitat i intuïtivitat de la interfície.

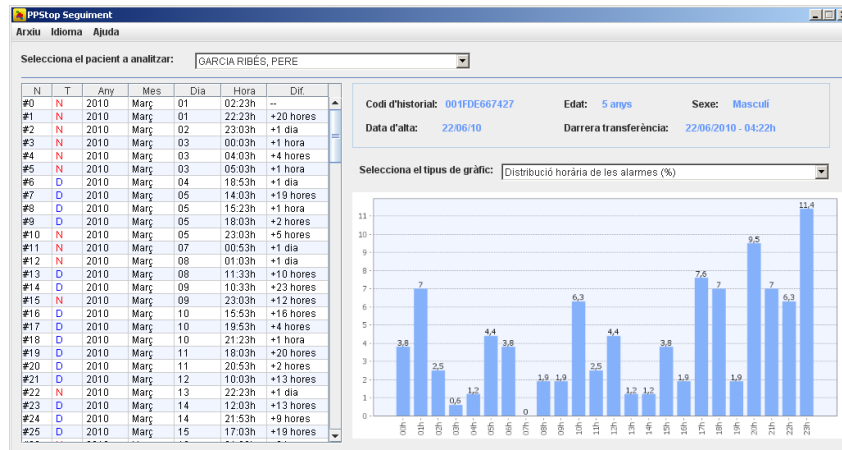


Figura 7.6: Captura de la pantalla principal de l'aplicació de seguiment

Tanmateix, també és interessant veure l'aparença de l'informe de seguiment PDF que genera l'aplicació, i que es pot veure en la figura 7.7. Es pot comprovar el resultat real de l'informe a l'annex D.5.

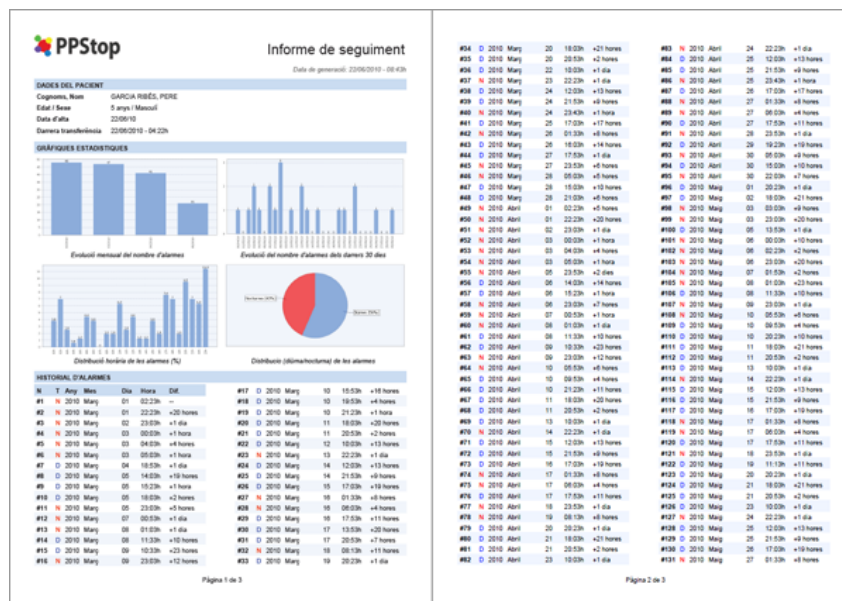


Figura 7.7: Captura d'un informe PDF generat per l'aplicació

## Capítol 8

# Preparació comercial

### 8.1 Plataforma de descàrrega per SMS

#### 8.1.1 WAP i la tecnologia WAP-Push

Wireless Application Protocol (WAP)[14] és un estàndar internacional, global i obert per a la transferència de dades a i des d'internet sobre una xarxa sense fils. Consisteix en un conjunt d'especificacions, definides per la *Open Mobile Alliance / WAP Forum* que defineixen un entorn d'aplicació i una pila de protocols per a accedir a aplicacions i serveis d'internet a través de terminals mòbils, típicament telèfons o Personal Digital Assistant (PDA). Cal tenir en compte que les xarxes de telefonia mòbil ofereixen unes prestacions generalment inferiors que els accessos convencionals a Internet, si bé és cert que amb l'aparició de les xarxes de 3a generació com UMTS les prestacions han millorat notablement.

La primera versió de *WAP* data de l'any 1998, i es caracteritza per tenir una pila de protocols pròpia no compatible directament amb internet. Aquesta incompatibilitat exigia la presència d'un node passarel·la per fer possible la intercomunicació entre un terminal *WAP* i un servidor de continguts *WAP* existent a internet, fet que en va frenar la seva expansió i va propiciar l'aparició d'una segona versió de *WAP* pocs anys més tard: *WAP 2.0*. La nova versió de *WAP* està present en tots els telèfons mòbils de nova generació (a partir de 2004), i és una reinvenió de *WAP* que utilitza llenguatges molt similars als emprats en la navegació en terminals no mòbils i que a més a més ha adoptat els protocols d'internet —Transmission Control Protocol (TCP) en la capa de transport i HTTP en la capa d'aplicació— amb opcions específiques per a millorar les prestacions d'aquests protocols sobre xarxes de comunicacions mòbils. Els mecanismes de seguretat s'han millorat i amb aquesta versió són compatibles amb els d'internet. Tot plegat fa que amb *WAP 2.0* la presència d'una passarel·la *WAP* no sigui estrictament necessària tot i que en certs casos pot aportar millores i funcions addicionals.

D'ençà a l'aparició de la tecnologia *WAP*, innevitablement n'han anat apareixent de relacionades per aportar funcions addicionals. Aquest és el cas de *WAP-Push*.

La tecnologia *WAP-Push* va ser incorporada en les especificacions de *WAP* a partir de la seva versió 1.2 per a permetre fer un "*push*" de contingut *WAP* en un terminal mòbil amb la mínima intervenció de l'usuari d'aquest terminal. Es tracta bàsicament d'un missatge especialment codificat que inclou un enllaç a una adreça *WAP*. El *WAP-Push* es situa a sobre del *WAP Datagram Protocol (WDP)* en la pila de protocols *WAP*, la qual cosa fa que pugui ser entregat per qualsevol portador que el suporti, com és el cas de *GPRS* o *SMS*. Malgrat això, generalment l'activació de *GPRS* des de la xarxa no està suportada amb la qual cosa els missatges *WAP Push* acostumen a ser distribuïts sobre *SMS*, opció més coneguda amb el nom de *SMS WAP-Push*. El procediment és el següent: quan un terminal rep un *SMS WAP-Push*, *WAP* dona automàticament a l'usuari l'opció de direccionar-lo cap a l'adreça especificada en el missatge, permetent així que l'usuari accedeixi directament al contingut de l'adreça des del seu terminal. Aquest contingut pot ser tant en format visual com en format fitxer cosa que n'habilita la seva descàrrega.

El cas més típic d'utilització de *WAP-Push* és la descàrrega de continguts multimèdia. Casos típics en són la descàrrega de melodies polifòniques, fons de pantalla, estalvis de pantalla, aplicacions *JAVA* per a mòbils, applets... o algunes campanyes publicitàries. La operativa de bona part d'aquests casos consisteix en l'enviament per part de l'usuari d'un *SMS* a un cert destinatari —una plataforma de descàrrega— indicant en el seu contingut un codi identificatiu de l'element multimèdia que volen descarregar-se. La plataforma de descàrrega interpreta el contingut del *SMS* i, en conseqüència, genera el missatge *WAP-Push* adient i l'envia a l'usuari. Quan aquest el rep, en obrir-lo *WAP* li ofereix la possibilitat d'accedir directament al contingut de l'adreça especificada pel missatge, i per tant, de descarregar-se'n el seu contingut. El benefici és mutu: l'usuari pot accedir d'una manera fàcil i còmoda a un contingut que desitja, i l'empresa encarregada de subministrar el missatge *WAP-Push* es queda amb una petita part de l'import que l'usuari final ha pagat per enviar el *SMS*.

### 8.1.2 Posada en funcionament

En el cas que ens ocupa, el nostre objectiu és utilitzar *WAP-Push* per a subministrar l'aplicació *J2ME* de *PPStop* a aquells clients que comprin el dispositiu *PPStop*. Si bé s'hagués pogut contractar el servei a una empresa externa que oferís serveis de *WAP-Push*, aquest s'ha implementat en un servidor *Ubuntu* anomenat *LEWIS* del Departament d'Enginyeria Telemàtica de la *UPC*.

fer-ho, s'ha provist a *LEWIS* d'un modem *GPRS* amb una tarjeta *SIM* amb número 44505 —des de la *UPC*—, i s'ha habilitat una plataforma de descàrrega per *SMS* fent ús de l'eina *SMS Server Tools*[15] en la seva versió 2.0. *SMS Server Tools* és un software programat en llenguatge *C* que permet l'enviament i la recepció de missatges curts a través de mòdems *GSM* i *GPRS*. Per a oferir el servei de *SMS WAP-Push*, s'ha generat

## 8.1. PLATAFORMA DE DESCÀRREGA PER SMS

un script (veure codis 8.1.2 i 8.1.2) programat en llenguatge Python[83] que s'encarrega del tractament dels missatges rebuts i de la generació i enviament dels missatges *WAP-Push*. Concretament aquest script realitza les següents tasques:

1. Defineix el proxy *ppstop* de descàrrega de *PPStop*.
2. Espera l'arribada d'un SMS.
3. Obté el número de telèfon d'origen i el contingut del missatge rebut.
4. Tracta el contingut del missatge i, si el contingut és *ppstop* l'associa al *proxy*.
5. Envia al destinatari un *SMS WAP-Push* amb l'enllaç cap al *proxy*.
6. Torna al punt 2.

El servidor de continguts que conté l'aplicació *PPStop* en aquest cas també està allotjat a *LEWIS*. Es tracta d'un servidor tradicional World Wide Web (WWW) que conté l'aplicació *PPStop* accessible via internet a través de la URL definida en el proxy de l'script anterior (<http://lewis.upc.es/ppstop/ppstop.jar>) per a que pugui ser descarregada al terminal. La figura 8.1 mostra l'escenari global i l'operativa global de funcionament de la plataforma de descàrrega.

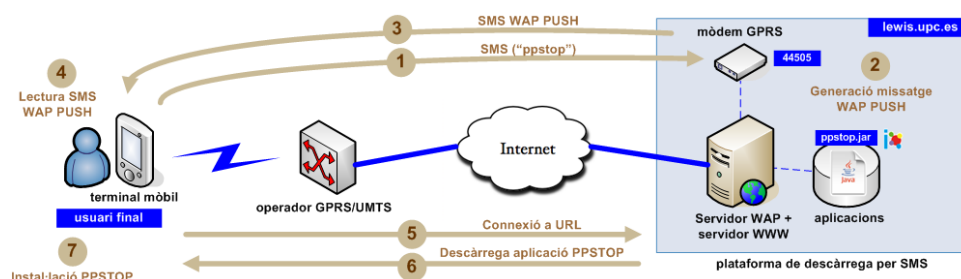


Figura 8.1: Plataforma de descàrrega WAP-Push per a *PPStop*

Com es pot apreciar, els passos globals des de l'enviament del SMS fins que l'usuari té instal·lat en el seu terminal l'aplicació *PPStop* són els següents:

1. L'usuari envia un SMS al número 44505 amb la paraula *ppstop* com a contingut.
2. La plataforma de descàrrega rep el SMS, l'interpreta i genera el missatge de resposta *WAP-Push*.
3. El *SMS WAP-Push* s'envia a l'usuari a través del mòdem *GPRS*.
4. L'usuari rep el SMS i l'accepta.
5. L'usuari és redireccionat cap a la URL del servidor de continguts de *LEWIS* que conté l'aplicació *PPStop*.
6. L'aplicació (*ppstop.jar*) es descarrega automàticament en el terminal de l'usuari.
7. L'usuari instal·la l'aplicació.

Codi 8.1: (SMSServer) SMSServer.py

```

1 #!/usr/bin/python
2
3 # A simple server for SMSs. This python script wraps the SMS Tools library
4 # to offer a simpler interface to SMS received events.
5 #
6 # (c) 2006, Juan Vera del Campo juanvi@entel.upc.es
7 # This script is under the GPL license
8 #
9 # This script uses the SMSTools
10 #
11
12 import string
13 import sys
14 import commands
15 import re
16
17 class SMSServer:
18     # Initializes the server with the device and (option) pin
19     def __init__(self, device, pin=None):
20         self.device=device
21         self.pin=pin
22         self.proxies={}
23         self.smsFormat=re.compile(r'.*^From: ([0-9+)]\n.*\n\n(.*)',re.S+re.M)
24     # sends an SMS (message) to a number
25     def send_SMS(self, number, message):
26         command=''
27         if self.pin:
28             command='putsms -p%s -d%s %s "%s"'%(self.pin,self.device,number,message)
29         else:
30             command='putsms -d%s %s "%s"'%(self.device,number,message)
31         print 'Running: ',command
32         (status,sms)=commands.getstatusoutput(command)
33         if status!=0: print '** Error: send status='+status
34     # gets an SMS and manages according to the registered proxies
35     def get_SMS(self):
36         # Waits for a SMS
37         command=''
38         if self.pin:
39             command='getsms -d%s -p%s'%(self.device,self.pin)
40         else:
41             command='getsms -d%s'%self.device
42         print 'Running: ',command
43         (status,sms)=commands.getstatusoutput(command)
44         if len(sms)==0:
45             return
46         else:
47             result=self.smsFormat.match(sms)
48             if result==None:
49                 print '** Error: Bad format for the SMS: ',sms
50                 return
51             number=result.group(1)
52             smscontent=result.group(2)
53             proxy=smscontent
54             content=smscontent
55             print '%s asks for %s'%(number,proxy)
56             f=None
57             # gets the proxy of the default manager, if any
58             try:
59                 f=self.proxies[proxy.upper()]
60             except:
61                 f=self.proxies['*']
62                 content=smscontent
63                 f(self,number,content)
64     # Adds a proxy.
65     # The proxy has the attributes SMSServer, number and content

```

## 8.2. WEB CORPORATIU DE PPSTOP

---

```
66 # The name * is for an special proxy: it gets the message when
67 # no other proxy matches the request, and gets the whole SMS
68 # message
69 def add_proxy(self,name,function):
70     print 'Proxy for %s: %s'%(name,function.__name__)
71     self.proxies[name.upper()]=function
72 # Runs the server. This method does not return.
73 def run(self):
74     try:
75         while True:
76             self.get_SMS()
77     except KeyboardInterrupt:
78         print 'Exiting'
```

Codi 8.2: (SMSServer) PPStop.py

```
1 #!/usr/bin/python
2
3 import SMSServer
4
5 # Definim el proxy de descàrrega del programa
6 def download_manager(server,number,content):
7     content=content.upper()
8     if content == 'PPSTOP':
9         server.send_SMS(number,'http://lewis.upc.es/ppstop/ppstop.jar')
10    else:
11        server.send_SMS(number,'Paraula no reconeguda...')
12
13 # Creem el servidor
14 server=SMSServer.SMSServer('/dev/ttyUSB0')
15 # Registrem el proxy
16 server.add_proxy('*',download_manager)
17 # Fem córrer el servidor
18 server.run()
```

## 8.2 Web corporatiu de PPStop

Per a donar a conèixer el producte en el mercat, la Fase 2 del projecte té planificada la realització d'una plana web informativa on s'especifiquin les característiques del producte, el seu funcionament bàsic, les seves avantatges respecte el sistema de tractament tradicional de l'enuresi, etc. És per això que ja se n'ha realitzat una primera versió (veure figura 8.3) que encara no s'ha fet pública i que actualment es troba allotjada en el servidor *LEWIS* del Departament: <http://lewis.upc.es/ppstop>.

La plana web s'ha realitzat amb Adobe Flash[84], tecnologia que permet la creació de contingut web dinàmic mitjançant animacions. Flash té ActionScript[85] com a llenguatge de programació, llenguatge que deriva del conegut JavaScript i que es caracteritza per ser orientat a objectes, on els objectes són els diferents elements de l'escenari de l'animació. A l'hora de fer el disseny de la pàgina s'han tingut en compte aspectes de temps de càrrega, resolució, conceptualitat i presentació de continguts, navegabilitat, accessibilitat i indexació per paraules clau. Per a il·lustrar gràficament la càrrega de la pàgina (620kB) s'ha afegit un precarregador a l'inici de la pàgina.

El web s'ha estructurat en quatre úniques seccions:

- **Pàgina principal:** Mostra la imatge del producte, els logotips corporatius, una enumeració d'idees principals a transmetre i dóna accés a les altres tres seccions que es descriuen a continuació.
- **Secció 'Què és PPStop':** Conté la informació relativa al funcionament global del sistema i del dispositiu PPStop, així com les millores que ofereix el sistema respecte el sistema de tractament tradicional.
- **Secció 'Aplicació de control':** Conté la informació relativa a les característiques i obtenció de l'aplicació de control del terminal mòbil, així com el seu manual d'usuari.
- **Secció 'Aplicació de seguiment':** Secció que conté informació relativa a les característiques i obtenció de l'aplicació de seguiment del tractament amb *PPStop*, així com el seu manual d'usuari.



Figura 8.2: Estructura de la web

En definitiva, la web permetrà que un possible client pugui percebre el potencial del producte i convèncer-lo per a què l'adquireixi.

## 8.3 Signatura del MIDlet

### 8.3.1 Problemàtica

La signatura d'una aplicació *J2ME* (o MIDlet) abans de comercialitzar-la és un aspecte a tenir molt en compte[86]. MIDP 2.0 té una arquitectura de seguretat que descriu vagament què han de fer els fabricants de telèfons mòbils per a complir-la. La idea bàsica és que durant l'execució d'un MIDlet en un terminal si aquest intenta accedir a APIs que poden suposar un risc de seguretat per a l'usuari —cas per exemple d'accedir als fitxers del terminal, de realitzar una connexió bluetooth, o fer ús de la càmera fotogràfica—, el terminal prèviament hauria de sol·licitar a l'usuari si dóna accés a l'aplicació per a utilitzar aquestes funcionalitats.

Aquesta prèvia sol·licitud que aparentment pot semblar comprensible esdevé un enorme maldecap per als programadors d'aplicacions per a terminals mòbils, sobretot en aquells casos en que aquestes funcionalitats s'utilitzen amb bastanta freqüència. Com que l'arquitectura de seguretat no descriu amb claredat el comportament a seguir, hi ha terminals



que fan aquesta sol·licitud tan sols una vegada per sessió —en l'inici de l'aplicació— però n'hi ha d'altres que ho fan cada vegada que l'aplicació intenta fer ús de la funcionalitat, interrompent sovint el correcte funcionament de l'aplicació a més a més de suposar això una incomoditat per a l'usuari de l'aplicació. El cas de l'aplicació de PPStop no n'és una excepció ja que fa un ús continu de les APIs JABWT (JSR-82) descrites en la secció 4.2.1 per a la creació d'un servei Bluetooth i l'atenció de les connexions Bluetooth procedents del dispositiu electrònic. En els models de terminal que demanen permís més d'una vegada per sessió el funcionament de l'aplicació PPStop es veu clarament perjudicada, ja que en cada intent de connexió —per exemple, en cas d'una connexió d'alarma— abans que aquesta pugui ser atesa per a així activar l'alarma, el terminal requereix que l'usuari doni permís per fer-ho. L'objectiu principal de l'aplicació que és la d'alarmar no s'assoleix, i per tant el sistema deixa de ser útil. Per a que això no succeeixi el que cal és certificar que l'ús de l'aplicació no suposa un risc de seguretat per a l'usuari, és a dir, cal signar el MIDlet.

#### 8.3.2 Signar un MIDlet

Signar un MIDlet no és una tasca senzilla[87][88] i malgrat la descripció del procés queda fora de l'abast d'aquest projecte, a continuació s'analitza a tot el que comporta i les repercussions econòmiques que pot arribar a tenir. Es tracta d'una tasca específica que depèn del dispositiu on s'ha d'executar el MIDlet —alguns dispositius en tenen prou amb que l'aplicació estigui signada per una entitat certificadora de confiança i d'altres necessiten que l'aplicació estigui signada pel propi fabricant— i que a més a més té un cost global considerable.

Un MIDlet que està signat certifica dos fets:

1. L'autor realment ha autoritzat l'ús del MIDlet.
2. El MIDlet es troba en el seu estat original, és a dir, no ha estat modificat fraudulentament per tercers.

Per a signar un MIDlet es fa ús de dos conceptes: la criptografia de clau pública —també coneguda com criptografia asimètrica[89]— i el concepte de certificat digital[90]. La cosa funciona aproximadament així: quan es signa una aplicació[91] se'n computa un *hash* de seguretat, i aquest és encriptat amb la clau privada per la persona o entitat que l'ha computat. Quan es vol verificar la signatura, el *hash* prèviament encriptat és desencriptat amb la clau pública de la persona o entitat que l'ha signat, i el verificador el computa novament a partir de l'aplicació que ha adquirit. Si el *hash* desencriptat i el computat concorden, la signatura s'ha verificat satisfactòriament. Per signar un MIDlet, doncs, només cal tenir en propietat una clau privada —qualsevol persona pot generar-ne una— per tant aquest procés podria portar-se a terme fàcilment per part del programador d'un MIDlet maliciós ja que podria autogenerar-se un parell de claus asimètriques —pública i privada— i autosignar-se ell mateix el seu MIDlet. Però en aquest cas la

signatura del MIDlet per certificar-ne la seva seguretat deixaria de tenir sentit. És per això que els diferents fabricants de terminals incorporen en aquests una llista d'entitats certificadores de confiança —entre les quals s'acostumen a trobar també els certificats de les operadores de telefonia mòbil que distribueixen els telèfons, per exemple *Vodafone*—, i només consideren de confiança els MIDlets signats per aquestes. I aquí és on comença a entrar en joc el tema de les repercussions econòmiques per part del programador ja que signar un MIDlet per una entitat certificadora de confiança reconeguda té un cost elevat.

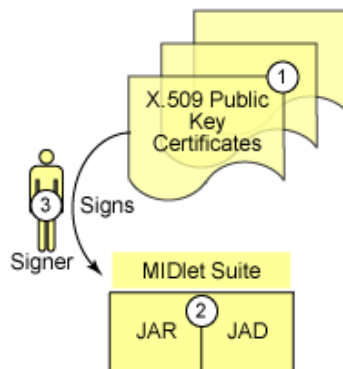


Figura 8.4: Signatura del MIDlet

Més enllà del cost de signar el MIDlet, també s'ha de considerar el cas en que el certificat de l'entitat certificadora, del fabricant o de l'operador utilitzat per a signar el MIDlet no sigui globalment reconegut com a entitat de confiança, malgrat sí que ho pugui ser per a bona part dels fabricants de terminals. En altres paraules, pot passar que els certificats de verificació d'aquestes entitats no estiguin disponibles en alguns terminals de fabricants que no el consideren de confiança, amb la qual cosa un MIDlet signat per una d'aquestes entitats no es pot verificar i per a aquells terminals l'aplicació és com si no hagués estat signada. Això implica que un desenvolupador de software que vulgués ampliar al màxim la compatibilitat de la seva aplicació es veu obligat a signar el MIDlet per més d'una entitat certificadora, incrementant encara més el cost d'aquest procés. Si encara així es vol tirar endavant la comercialització del producte, encara cal tenir en compte un parell de detalls més. Primerament, la signatura d'un MIDlet té caducitat, és a dir, aquesta només és vàlida durant un període de temps; passat aquest període el certificat de l'entitat certificadora caduca i cal renovar el certificat i tornar a signar el MIDlet per mantenir operativa l'aplicació provocant un cost econòmic com a mínim lineal en el temps. En segon lloc el cost de signar una aplicació acostuma a ser per MIDlet, per model de terminal i per intent de signat. Això vol dir que per cada canvi que es faci a l'aplicació una vegada signada —per exemple, una actualització de versió del software—, per cada vegada que aquest s'envii a signar i no passi les proves de seguretat requerides per l'entitat certificadora o per cada model de terminal sobre el qual es vulgui fer funcionar el MIDlet, implica un nou procés de signat i el conseqüent pagament a l'entitat certificadora.

Per posar un exemple real, signar un MIDlet amb *CapGemini* —una entitat certificadora que acostuma a aparèixer en les llistes d'entitats certificadores de confiança dels terminals— costa uns 240 euros per al primer intent i 210 euros per a cadascun dels intents següents al primer intent fallit. I així se'n podrien analitzar moltes altres de conegudes com ara *Thawte*, *Verisign*, *GoDaddy*...

Amb tot això, es just i comprensible que en el mercat el nombre d'aplicacions signades sigui molt reduït ja que suposa una barrera molt important a nivell econòmic per als programadors d'aplicacions, i l'ús de la tecnologia *J2ME* per a la programació d'aplicacions amb finalitats comercials per a terminals mòbils en certa manera se'n veu força resentida[92].

**PPStop** | Sistema inalàmbic de control i seguiment de l'enuresi

Escull una opció per ampliar la informació

**Què és PPStop?**

PPStop és un dispositiu inalàmbic que millora el sistema tradicional de tractament de l'enuresi

- 1 SENSE CABLEJAT
- 2 CONNECTIVITAT BLUETOOTH
- 3 DIMENSIONS REDUÏDES
- 4 BATERIA DE LLARGA DURADA
- 5 SENSE RISCOS PER A LA SALUT

**Aplicació de control**

Controla el tractament de l'enuresi des del terminal mòbil

- ✓ INTERFICIE MULTIDIOMA
- ✓ ALARMA SONORA
- ✓ SENSIBILITAT CONFIGURABLE
- ✓ CONTROL DE BATERIA
- ✓ REGISTRE D'ALARMES
- ✓ TRANSMISSIÓ D'ESTADÍSTIQUES

**Aplicació de seguiment**

Segueix l'evolució del tractament analitzant-ne les estadístiques

- ✓ MÚLTIPLES DISPOSITIUS
- ✓ GRÀFICS D'EVOLUCIÓ
- ✓ GENERACIÓ D'INFORMES
- ✓ LLEUGER I VERSÀTIL

COMPATIBLE

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH | entel | hospital sant joan de déu | innova

audio | ON

(a) Homepage

**PPStop** | Sistema inalàmbic de control i seguiment de l'enuresi

**Què és PPStop?**

**Un sistema innovador**

PPStop és un dispositiu inalàmbic que millora el tractament tradicional de l'enuresi.

El tractament tradicional consisteix en un sensor d'humitat connectat de manera cablejada a un timbre que el pacient porta penjat en un lloc visible i de fàcil accés. Quan el pacient s'orina, el sensor ho detecta, activant el timbre del pacient. En ser un sistema cablejat, provoca molèsties en el pacient, alhora que el timbre no passa desapercbut i en vulnera la seva intimitat.

PPStop millora el sistema en aquests dos sentits. El dispositiu PPStop a més a més del sensor d'humitat incorpora un transmissor bluetooth que genera les alertes mitjançant connectivitat bluetooth a l'aplicació PPStop: una aplicació que s'executa en el propi terminal mòbil del pacient.

**Més funcionalitats**

La connectivitat bluetooth de PPStop permet que les alertes no només siguin generades en el terminal del pacient, si no que també poden ser generades a tercers persones (supervisors) que es trobin dins del rang de cobertura del dispositiu.

L'aplicació PPStop permet gestionar en tot moment els usuaris associats al dispositiu i també monitoritzar i configurar paràmetres com ara l'estat de la bateria o la sensibilitat del sensor d'humitat. A més a més, el sistema ofereix la possibilitat de transmetre les estadístiques d'alarmes a un programa de seguiment de PPStop que s'executa en un ordinador personal.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH | entel | hospital sant joan de déu | innova

audio | ON

(b) Secció "Què és PPStop?"

**PPStop** | Sistema inalàmbic de control i seguiment de l'enuresi

**Aplicació de control**

**Les alarmes, ara al teu mòbil!**

L'aplicació de control PPStop s'executa en un terminal mòbil i és l'encarregada de generar l'avís sonor quan rep una alarma del dispositiu PPStop. És compatible amb tots els terminals que suportin JAVA i disposin de connectivitat bluetooth.

Mitjançant aquesta aplicació es poden monitoritzar i controlar alguns paràmetres del dispositiu PPStop, com ara l'estat de la bateria, canviar la sensibilitat del sensor d'humitat, associar nous terminals al dispositiu, consultar l'historial d'alarmes, etc.

Aquestes últimes, a més a més, poden ser transferides per bluetooth a l'aplicació de seguiment PPStop en cas que es vulgui fer un estudi detallat del seguiment del tractament.

**Com obtenir l'aplicació?**

Per obtenir l'aplicació de control has d'enviar un sms amb la paraula "ppstop". Automàticament rebràs l'aplicació al teu terminal via WAP Push.

**ENVIJA ppstop AL 44505**

TERMINALS COMPATIBLES  
MANUAL DE L'USUARI

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH | entel | hospital sant joan de déu | innova

audio | ON

(c) Secció "Aplicació de control"

Figura 8.3: Captures de la pàgina web corporativa de PPStop

## Capítol 9

# Conclusions i línies futures

### 9.1 Conclusions

L'objectiu inicial i principal del projecte era l'obtenció d'un prototipus funcional de dispositiu electrònic que a partir de la captació d'un event extern aconseguís activar —mitjançant bluetooth— una alarma sonora en un terminal mòbil dotat d'una aplicació especialment programada per a aquest efecte, en aquest cas una aplicació programada en J2ME. Després d'haver obtingut el primer prototip funcional, d'haver tingut en consideració tots els elements incidents en el sistema global i de veure els resultats obtinguts, es pot concloure que l'objectiu principal del projecte s'ha assolit amb escreix. S'ha aconseguit fer complir tots els requisits que el client va demanar inicialment en la fase de presa de requisits —tant els tècnics com els funcionals— demostrant que la fase de disseny del sistema va fer-se amb encert. S'han realitzat proves pilot per testejar el funcionament i la utilitat del sistema als laboratoris de recerca de l'Hospital Sant Joan de Déu, on s'ha pogut comprovar que aquesta primera versió del producte millora el sistema tradicional de tractament de l'enuresi i que, per tant, pot ser vàlida per a la comercialització. Adicionalment, també s'ha assolit amb èxit la realització d'una aplicació per a PC que permetés realitzar el seguiment del tractament de l'enuresi en aquells pacients que utilitzen el sistema PPstop: l'aplicació adquireix correctament les alarmes enregistrades per l'aplicació del terminal mòbil i genera estadístiques i informes útils per al seguiment mèdic del tractament.

S'han utilitzat tecnologies cada vegada més populars i acceptades en el món de les comunicacions PAN i la programació per a terminals mòbils —Bluetooth i J2ME, respectivament—. Tant és així que el nombre de fabricants que han afegit compatibilitat per a aquestes tecnologies en els seus terminals mòbils ha crescut fins al punt que actualment és difícil trobar al mercat un terminal sense connectivitat Bluetooth o sense suport per a J2ME. Això es pot comprovar fàcilment consultant les característiques tècniques dels models dels principals fabricants del sector —*Nokia, Sony Ericsson, Samsung, Motorola, LG*— actualment en el mercat. Ara per ara, la combinació J2ME+Bluetooth

—també J2SE+Bluetooth— està molt treballada i l'existència d'APIs com *JABWT* en faciliten molt el seu ús, obrint moltes possibilitats de mercat en la creació de nous serveis per a l'usuari. Segurament —com ja s'ha comentat en el capítol 8— una de les úniques barreres existents en aquest sentit és la repercussió econòmica que té el signat de MIDlets. No obstant, dins de la comunitat J2ME hi ha unanimitat en que cal millorar en aquest sentit; i en la versió 3.0 de MIDP —aprovada el passat mes de desembre de 2009— ja s'hi consideren algunes millores.

A nivell de disseny, un dels punts més crítics que ha calgut superar ha estat el dissenyar el protocol de comunicació entre dispositiu electrònic i terminal tenint en compte dos condicionants bàsics per al correcte funcionament del sistema. El primer, que el temps de resposta d'alarma per al primer terminal alertat no podia superar els 3 segons —temps límit que els experts en enuresi consideren que és útil alarmar al pacient— i el segon, que el temps de vida del dispositiu electrònic havia de ser suficientment llarg com per fer-ne viable la comercialització. Donada la durada d'alguns protocols existents de Bluetooth com ara DDP o SDP i donat el considerable consum de Bluetooth en transmissió o escolta de connexions, ha calgut redissenyar tot el protocol. No ha estat fàcil aconseguir un temps de resposta baix i ha calgut renunciar a l'utilització de DDP+SDP en cada procés d'alarma. També ha calgut renunciar a l'intercanvi de rols master-esclau entre el dispositiu electrònic i el terminal per ser inviable l'ús del dispositiu electrònic com a esclau per l'elevat consum de Bluetooth, i que tant bé hagués anat per a l'enviament d'ordres de configuració al dispositiu electrònic des de l'aplicació del terminal. Tot plegat ha fet que el protocol dissenyat en el capítol 3 hagi esdevingut un tant *atípic* respecte el que hagués pogut ser en circumstàncies normals.

Un altre dels punts que ha calgut parar més atenció i que conseqüentment ha requerit una major dedicació respecte la resta d'elements del sistema ha estat en la programació del microcontrolador. Tot i haver realitzat una simulació de la unitat de control per a comprovar la correcta presa de decisions per al seguiment del protocol dissenyat, la migració de l'aplicació del simulador JAVA al llenguatge C no ha estat fàcil. C no té tantes llibreries com JAVA i ha calgut implementar manualment moltes de les funcionalitats —per exemple, la realització d'una cua circular—. No obstant, el principal problema ha esdevingut la limitada capacitat de memòria —únicament 4kB— per a codi de programa. S'ha hagut de treballar de valent per reduir el nombre de llibreries utilitzades, simplificar al màxim el codi i reduir l'ús de variables. Altres problemes com la sincronització entre diverses lectures i escriptures entre microcontrolador i transmissor bluetooth o la gestió de les interrupcions i del mode *power-down* també ha calgut resoldre'ls, amb la dificultat afegida de no poder utilitzar les avantatges d'un mode de *debug* que permetés veure exactament què és el que estava fallant en cada moment. Tot plegat ha fet que la planificació de tasques feta a l'inici del projecte —capítol 1— s'hagi vist lleugerament alterada per aquests colls d'ampolla. No obstant, finalment s'han pogut superar aquestes dificultats i s'ha aconseguit que el microcontrolador funcionés correctament.

Experimentalment i tal com s'ha descrit en el capítol 6, s'han mesurat els temps de resposta d'alarma per als diferents escenaris reals en què es pot trobar el sistema. Això

ha permès determinar-ne el seu comportament. Per exemple, s'ha observat com el temps de resposta d'una connexió Bluetooth es veu perjudicat per la distància, del nombre d'obstacles materials que es trobin enmig —persones, parets, etc— i de les interferències que puguin existir en la banda de freqüències on opera Bluetooth, degut a que Bluetooth és una tecnologia que s'adapta a les condicions de l'entorn i en els canals menys fiables realitza transmissions més robustes a canvi de reduir les taxes útils de transmissió i recepció. Com que el transmissor Bluetooth opera a baixa potència per a així minimitzar el consum energètic, el rang de cobertura Bluetooth s'ha vist reduït del que inicialment s'estimava que podria aconseguir. No obstant, els experiments han demostrat el correcte funcionament del sistema per a qualsevol situació en un rang de cobertura màxim d'uns 15-20 metres.

És interessant remarcar també l'ús que s'ha fet del patró de programació model-vista-controlador i l'ús d'APIs molt professionalitzades com *LWUIT* vistos en el capítol 4 per a la creació de l'entorn gràfic de l'aplicació de control, que han permès obtenir un programa ben estructurat des del punt de vista del programador i d'alta qualitat des del punt de vista gràfic. Això suposa un avantatge si es té una visió de futur del projecte. Iniciar una programació professionalitzada i ben estructurada des d'un bon principi garanteix que la implementació de pròximes versions de l'aplicació siguin més fàcils de dur a terme i el número de mòduls a canviar en cas de ser necessari fer modificacions serà mínim.

A nivell de funcionalitat, s'han incorporat totes aquelles que eren requerides i més, pensant sempre en una futura comercialització del producte. Funcionalitats no contemplades inicialment com ara el suport multi-idioma de l'aplicació del terminal —ampliant així el target del producte—, la transferència d'historial d'alarmes a l'aplicació de seguiment del tractament o la possibilitat d'escollir el tipus d'alarma —per exemple, veu humana per al pacient i ring convencional per a un supervisor— s'han incorporat amb l'objectiu de millorar les prestacions globals del sistema.

En resum, es pot dir que s'han complert els objectius, els resultats han estat els esperats i que aquests són satisfactoris per a l'ús que se li vol donar al sistema.

## 9.2 Línies futures

Si bé és cert que els objectius inicialment plantejats s'han aconseguit, el projecte PPStop és un projecte que té plantejades 3 fases i de les quals actualment només se n'han assolit les dos primeres. El futur del projecte passa per implementar la fase 3. Aquesta es basa fonamentalment en el pas de prototip a producte per a la seva posterior comercialització, i alguns dels passos a realitzar ja s'han comentat i plantejat en el capítol 8: posar a la xarxa i promocionar el web corporatiu del producte PPStop; la realització de més proves pilot amb pacients, presa de resultats i millora continua del sistema en aquells aspectes que calgui; i la signatura del MIDlet definitiu per a la seva comercialització.

No obstant, també hi ha una important línia futura de treball per part de la casa

d'electrònica encarregada de la fabricació del dispositiu electrònic —*Electrònica Feixas*— per a tractar de millorar aspectes respecte aquesta primera versió del prototip. Una d'elles seria la incorporació d'una bateria de major capacitat nominal per a allargar el temps de vida del dispositiu. La segona és incorporar en el dispositiu electrònic el circuit detector del nivell real de la bateria. Actualment el protocol porta suport per a aquesta acció, però encara no ha estat implementada per la casa d'electrònica amb la qual cosa ara per ara el microcontrolador proporciona un valor fix per a aquest camp. També seria molt interessant el redisseny de l'encapsulat del dispositiu B.6, fent que fos més homogeni i amb un millor enresinat i un millor acabat.

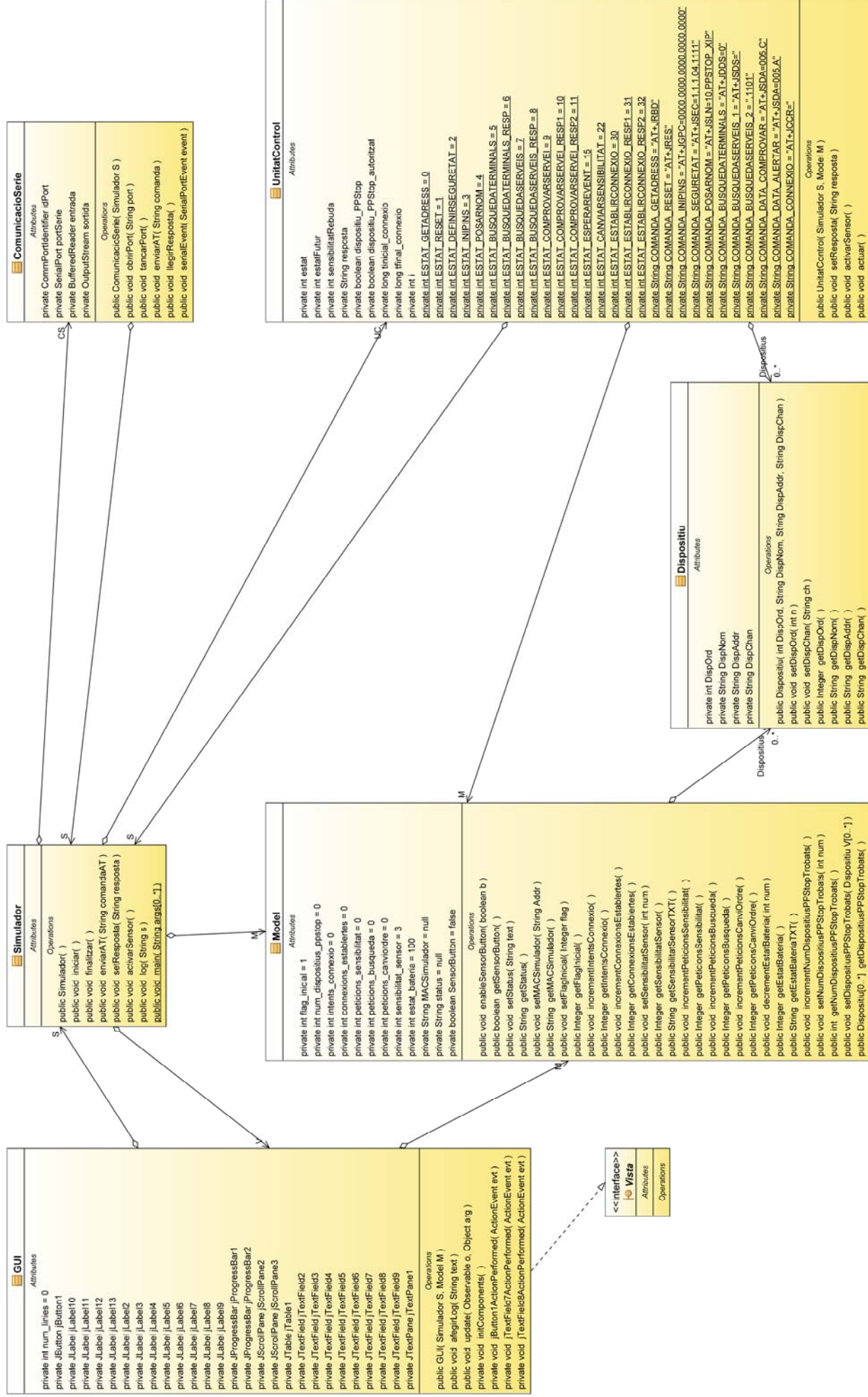


## Apèndix A

### Simulador del dispositiu electrònic



## A.1 Diagrama de classes del simulador



## A.2 JavaDoc del simulador

---

**Package**

**ppstop\_simulador**

## ppstop\_simulador

### Class ComunicacioSerie

java.lang.Object

└─ppstop\_simulador.ComunicacioSerie

#### All Implemented Interfaces:

javax.comm.SerialPortEventListener

```
public class ComunicacioSerie
extends java.lang.Object
implements javax.comm.SerialPortEventListener
```

Classe per a la Comunicació Sèrie amb el transmissor BT

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

1.0 (Juliol de 2009)

### Field Summary

private	<a href="#">entrada</a> Bufere de recepció de dades
private	<a href="#">idPort</a> Identificador del port sèrie
private	<a href="#">portSerie</a> Objecte de control del Port Sèrie
private	<a href="#">S</a> Referència al simulador
private	<a href="#">sortida</a> Stream de sortida de dades

### Constructor Summary

public	<a href="#">ComunicacioSerie</a> ( <a href="#">Simulador</a> S) Crea una instància de ComunicacióSerie
--------	---

### Method Summary

void	<a href="#">enviarAT</a> (java.lang.String comanda) Envia una comanda AT al transmissor BT
void	<a href="#">llegirResposta</a> () Llegeix la resposta del transmissor BT i ho notifica al simulador
void	<a href="#">obrirPort</a> (java.lang.String port) Obre una comunicació per un port sèrie determinat
void	<a href="#">serialEvent</a> (javax.comm.SerialPortEvent event) Detecta un event en el Port Sèrie i actua en conseqüència

void	<a href="#">tancarPort()</a> Tanca la comunicació pel port sèrie
------	---

#### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`, `toString`, `wait`, `wait`, `wait`

#### Methods inherited from interface `javax.comm.SerialPortEventListener`

`serialEvent`

## Fields

### S

`private ppstop_simulador.Simulador S`

Referència al simulador

### idPort

`private javax.comm.CommPortIdentifier idPort`

Identificador del port sèrie

### portSerie

`private javax.comm.SerialPort portSerie`

Objecte de control del Port Sèrie

### entrada

`private java.io.BufferedReader entrada`

Bufer de recepció de dades

### sortida

`private java.io.OutputStream sortida`

Stream de sortida de dades

## Constructors

### ComunicacioSerie

`public ComunicacioSerie(Simulador S)`

Crea una instància de ComunicacióSerie

#### Parameters:

S - Referència al simulador

(continued on next page)



(continued from last page)

## Methods

### **obrirPort**

```
public void obrirPort(java.lang.String port)
```

Obre una comunicació per un port sèrie determinat

**Parameters:**

port - Identificador del port a obrir

---

### **tancarPort**

```
public void tancarPort()
```

Tanca la comunicació pel port sèrie

---

### **enviarAT**

```
public void enviarAT(java.lang.String comanda)
```

Envia una comanda AT al transmissor BT

**Parameters:**

comanda - Comanda AT a enviar

---

### **llegirResposta**

```
public void llegirResposta()
```

Llegeix la resposta del transmissor BT i ho notifica al simulador

---

### **serialEvent**

```
public void serialEvent(javax.comm.SerialPortEvent event)
```

Detecta un event en el Port Sèrie i actua en conseqüència

**Parameters:**

event - Event detectat

---

# ppstop\_simulador

## Class Dispositiu

java.lang.Object

└─ppstop\_simulador.Dispositiu

public class **Dispositiu**  
extends java.lang.Object

Classe representativa d'un dispositiu bluetooth (terminal mòbil PPStop)

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juliol de 2009)

### Field Summary

private	<a href="#">DispAddr</a> Adreça bluetooth del dispositiu
private	<a href="#">DispChan</a> Canal del servei PPStop del dispositiu
private	<a href="#">DispNom</a> Nom del dispositiu
private	<a href="#">DispOrd</a> Ordre del dispositiu

### Constructor Summary

public	<a href="#">Dispositiu</a> (int DispOrd, java.lang.String DispNom, java.lang.String DispAddr, java.lang.String DispChan) Crea una instància de Dispositiu
--------	--

### Method Summary

java.lang.String	<a href="#">getDispAddr</a> () Obté l'adreça bluetooth del dispositiu
java.lang.String	<a href="#">getDispChan</a> () Obté el canal del servei PPStop del dispositiu
java.lang.String	<a href="#">getDispNom</a> () Obté el nom del dispositiu
java.lang.Integer	<a href="#">getDispOrd</a> () Obté l'ordre del dispositiu
void	<a href="#">setDispChan</a> (java.lang.String ch) Estableix el canal del dispositiu
void	<a href="#">setDispOrd</a> (int n) Estableix l'ordre del dispositiu

**Methods inherited from class** `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait`

## Fields

### DispOrd

```
private int DispOrd
```

Ordre del dispositiu

### DispNom

```
private java.lang.String DispNom
```

Nom del dispositiu

### DispAddr

```
private java.lang.String DispAddr
```

Adreça bluetooth del dispositiu

### DispChan

```
private java.lang.String DispChan
```

Canal del servei PPStop del dispositiu

## Constructors

### Dispositiu

```
public Dispositiu(int DispOrd,  
                  java.lang.String DispNom,  
                  java.lang.String DispAddr,  
                  java.lang.String DispChan)
```

Crea una instància de Dispositiu

**Parameters:**

DispOrd - Ordre del dispositiu

DispNom - Nom del dispositiu

DispAddr - Adreça bluetooth del dispositiu

DispChan - Canal del servei PPStop del dispositiu

## Methods

### setDispOrd

```
public void setDispOrd(int n)
```

Estableix l'ordre del dispositiu

(continued from last page)

**Parameters:**

n - Ordre del dispositiu

---

**setDispChan**

```
public void setDispChan(java.lang.String ch)
```

Estableix el canal del dispositiu

**Parameters:**

ch - Canal del dispositiu

---

**getDispOrd**

```
public java.lang.Integer getDispOrd()
```

Obté l'ordre del dispositiu

**Returns:**

Ordre del dispositiu

---

**getDispNom**

```
public java.lang.String getDispNom()
```

Obté el nom del dispositiu

**Returns:**

Nom del dispositiu

---

**getDispAddr**

```
public java.lang.String getDispAddr()
```

Obté l'adreça bluetooth del dispositiu

**Returns:**

Adreça bluetooth del dispositiu

---

**getDispChan**

```
public java.lang.String getDispChan()
```

Obté el canal del servei PPStop del dispositiu

**Returns:**

Número de canal del servei PPStop

## ppstop\_simulador Class GUI

```

java.lang.Object
|
+- java.awt.Component
|   |
|   +- java.awt.Container
|       |
|       +- java.awt.Window
|           |
|           +- java.awt.Frame
|               |
|               +- javax.swing.JFrame
|                   |
|                   +- ppstop_simulador.GUI

```

### All Implemented Interfaces:

[Vista](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible, java.awt.MenuContainer, javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.RootPaneContainer, javax.accessibility.Accessible, javax.swing.WindowConstants

### public class GUI

extends javax.swing.JFrame

implements javax.swing.WindowConstants, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler, java.awt.MenuContainer, javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, [Vista](#)

Clase que genera la Graphic User Interface (GUI) de la Vista del simulador (patr  MVC)

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

1.0 (Juliol de 2009)

Field Summary		
private		<a href="#">jButton1</a>
private		<a href="#">jLabel10</a>
private		<a href="#">jLabel11</a>
private		<a href="#">jLabel12</a>
private		<a href="#">jLabel13</a>
private		<a href="#">jLabel2</a>
private		<a href="#">jLabel3</a>
private		<a href="#">jLabel4</a>
private		<a href="#">jLabel5</a>

private	<a href="#">jLabel6</a>
private	<a href="#">jLabel7</a>
private	<a href="#">jLabel8</a>
private	<a href="#">jLabel9</a>
private	<a href="#">jProgressBar1</a>
private	<a href="#">jProgressBar2</a>
private	<a href="#">jScrollPane2</a>
private	<a href="#">jScrollPane3</a>
private	<a href="#">jTable1</a>
private	<a href="#">jTextField2</a>
private	<a href="#">jTextField3</a>
private	<a href="#">jTextField4</a>
private	<a href="#">jTextField5</a>
private	<a href="#">jTextField6</a>
private	<a href="#">jTextField7</a>
private	<a href="#">jTextField8</a>
private	<a href="#">jTextField9</a>
private	<a href="#">jTextPanel</a>
private	<a href="#">M</a> Referència al model de dades
private	<a href="#">num_linies</a> Nombre de línies de la finestra de LOG
private	<a href="#">S</a> Referència al Simulador

#### Fields inherited from class `javax.swing.JFrame`

`accessibleContext`, `defaultCloseOperation`, `defaultLookAndFeelDecoratedKey`, `EXIT_ON_CLOSE`, `rootPane`, `rootPaneCheckingEnabled`, `transferHandler`

#### Fields inherited from class `java.awt.Frame`

base, CROSSHAIR\_CURSOR, DEFAULT\_CURSOR, E\_RESIZE\_CURSOR, frameSerializedDataVersion, HAND\_CURSOR, ICONIFIED, MAXIMIZED\_BOTH, MAXIMIZED\_HORIZ, MAXIMIZED\_VERT, maximizedBounds, mbManagement, menuBar, MOVE\_CURSOR, N\_RESIZE\_CURSOR, nameCounter, NE\_RESIZE\_CURSOR, NORMAL, NW\_RESIZE\_CURSOR, ownedWindows, resizable, S\_RESIZE\_CURSOR, SE\_RESIZE\_CURSOR, serialVersionUID, state, SW\_RESIZE\_CURSOR, TEXT\_CURSOR, title, undecorated, W\_RESIZE\_CURSOR, WAIT\_CURSOR

#### Fields inherited from class `java.awt.Window`

allWindows, alwaysOnTop, anchor, base, beforeFirstShow, beforeFirstWindowShown, focusableWindowState, focusMgr, icons, inputContext, inputContextLock, isInShow, isTrayIconWindow, locationByPlatform, locationByPlatformProp, log, modalBlocker, modalExclusionType, nameCounter, opacity, opaque, OPENED, ownedWindowList, securityWarningAlignmentX, securityWarningAlignmentY, securityWarningHeight, securityWarningPointX, securityWarningPointY, securityWarningWidth, serialVersionUID, shape, showWithParent, state, syncLWRequests, systemSyncLWRequests, temporaryLostComponent, TRANSPARENT\_BACKGROUND\_COLOR, warningString, weakThis, windowFocusListener, windowListener, windowSerializedDataVersion, windowStateListener

#### Fields inherited from class `java.awt.Container`

component, containerListener, containerSerializedDataVersion, dbg, descendantsCount, dispatcher, EMPTY\_ARRAY, focusCycleRoot, focusTraversalPolicy, focusTraversalPolicyProvider, INCLUDE\_SELF, layoutMgr, listeningBoundsChildren, listeningChildren, mixingLog, modalAppContext, modalComp, numOfHWComponents, numOfLWComponents, preserveBackgroundColor, printing, printingThreads, SEARCH\_HEAVYWEIGHTS, serialPersistentFields, serialVersionUID

#### Fields inherited from class `java.awt.Component`

accessibleContext, actionListenerK, adjustmentListenerK, appContext, background, backgroundEraseDisabled, BOTTOM\_ALIGNMENT, boundsOp, bufferStrategy, CENTER\_ALIGNMENT, changeSupport, coalesceEventsParams, coalesceMap, coalescingEnabled, componentListener, componentListenerK, componentOrientation, componentSerializedDataVersion, compoundShape, containerListenerK, cursor, dbg, dropTarget, enabled, eventCache, eventMask, FOCUS\_TRAVERSABLE\_DEFAULT, FOCUS\_TRAVERSABLE\_SET, FOCUS\_TRAVERSABLE\_UNKNOWN, focusable, focusListener, focusListenerK, focusLog, focusTraversalKeyPropertyNames, focusTraversalKeys, focusTraversalKeysEnabled, font, foreground, graphicsConfig, height, hierarchyBoundsListener, hierarchyBoundsListenerK, hierarchyListener, hierarchyListenerK, ignoreRepaint, incRate, inputMethodListener, inputMethodListenerK, isAddNotifyComplete, isFocusTraversableOverridden, isInc, isOpaqueForMixing, isPacked, itemListenerK, keyListener, keyListenerK, LEFT\_ALIGNMENT, locale, LOCK, log, maxSize, maxSizeSet, minSize, minSizeSet, mixingLog, mouseListener, mouseListenerK, mouseMotionListener, mouseMotionListenerK, mouseWheelListener, mouseWheelListenerK, name, nameExplicitlySet, nativeInLightFixer, newEventsOnly, objectLock, opaquePropertyChangeListener, ownedWindowK, parent, peer, peerFont, popups, prefSize, prefSizeSet, privateKey, requestFocusController, RIGHT\_ALIGNMENT, serialVersionUID, textListenerK, TOP\_ALIGNMENT, valid, visible, width, windowClosingException, windowFocusListenerK, windowListenerK, windowStateListenerK, x, y

#### Fields inherited from interface `java.awt.image.ImageObserver`

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

#### Fields inherited from interface `javax.swing.WindowConstants`

DISPOSE\_ON\_CLOSE, DO\_NOTHING\_ON\_CLOSE, EXIT\_ON\_CLOSE, HIDE\_ON\_CLOSE

## Constructor Summary

public	<a href="#">GUI</a> ( <a href="#">Simulador</a> S, <a href="#">Model</a> M) Crea una nova instància de GUI
--------	---

## Method Summary

void	<a href="#">afegirLog</a> (java.lang.String text) Afegeix una línia de text a la finestra de LOG
void	<a href="#">initComponents</a> () Inicialitza els components de la GUI
void	<a href="#">jButton1ActionPerformed</a> (java.awt.event.ActionEvent evt) Acció d'activar el sensor d'humitat en polsar el botó de la GUI
void	<a href="#">jTextField7ActionPerformed</a> (java.awt.event.ActionEvent evt)
void	<a href="#">jTextField8ActionPerformed</a> (java.awt.event.ActionEvent evt)
void	<a href="#">update</a> (java.util.Observable o, java.lang.Object arg) Actualitza la vista amb les noves dades del Model (patró MVC)

### Methods inherited from class javax.swing.JFrame

addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update

### Methods inherited from class java.awt.Frame

addNotify, constructComponentName, getAccessibleContext, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, init, initIDs, isFrameStateSupported, isResizable, isUndecorated, paramString, postProcessKeyEvent, readObject, remove, removeNotify, setCursor, setExtendedState, setIconImage, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated, writeObject

### Methods inherited from class java.awt.Window



```
addNotify, addOwnedWindow, addPropertyChangeListener, addPropertyChangeListener,
addToWindowList, addWindowFocusListener, addWindowListener, addWindowStateListener,
adjustDecendantsOnParent, adjustListeningChildrenOnParent, applyCompoundShape,
applyCurrentShape, applyOpacity, applyOpaque, applyResourceBundle,
applyResourceBundle, applyShape, calculateSecurityWarningPosition,
canContainFocusOwner, clearMostRecentFocusOwnerOnHide, connectOwnedWindow,
constructComponentName, createBufferStrategy, createBufferStrategy,
deliverMouseWheelToAncestor, deserializeResources, dispatchEventImpl,
dispatchMouseWheelToAncestor, dispose, disposeImpl, doDispose, doesClassImplement,
doesImplement, eventEnabled, getAccessibleContext, getAllUnblockedWindows,
getAllWindows, getBufferStrategy, getContainer, getDocumentRoot,
getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner,
getFocusTraversalKeys, getGraphicsConfiguration, getIconImages, getInputContext,
getListeners, getLocale, getLocationOnWindow, getModalBlocker, getModalExclusionType,
getMostRecentFocusOwner, getOwnedWindows_NoClientCode, getOwnedWindows,
getOwner_NoClientCode, getOwner, getOwnerlessWindows, getTemporaryLostComponent,
getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners,
getWindows, getWindows, getWindowStateListeners, hide, init, initDeserializedWindow,
initIDs, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isFocusableWindow,
isFocusCycleRoot, isFocused, isLocationByPlatform, isModalBlocked, isModalExcluded,
isRecursivelyVisible, isShowing, mixOnReshaping, ownedInit, pack, postEvent,
postProcessKeyEvent, postWindowEvent, preProcessKeyEvent, processEvent,
processWindowEvent, processWindowFocusEvent, processWindowStateEvent, readObject,
removeFromWindowList, removeFromWindowList, removeNotify, removeOwnedWindow,
removeWindowFocusListener, removeWindowListener, removeWindowStateListener, resetGC,
reshape, setAlwaysOnTop, setBounds, setBounds, setClientSize, setCursor,
setFocusableWindowState, setFocusCycleRoot, setIconImage, setIconImages,
setLayersOpaque, setLocationByPlatform, setLocationRelativeTo, setMinimumSize,
setModalBlocked, setModalExclusionType, setSize, setSize, setTemporaryLostComponent,
setVisible, setWarningString, show, toBack_NoClientCode, toBack,
toFront_NoClientCode, toFront, updateChildFocusableWindowState,
updateChildrenBlocking, updateWindow, writeObject
```

**Methods inherited from class** java.awt.Container

```

add, add, add, add, add, addContainerListener, addDelicately, addImpl, addNotify,
addPropertyChangeListener, addPropertyChangeListener, adjustDecendantsOnParent,
adjustDescendants, adjustListeningChildren, applyComponentOrientation,
areFocusTraversalKeysSet, canContainFocusOwner, checkAdding, checkAddToSelf, checkGD,
checkNotAWindow, clearCurrentFocusCycleRootOnHide, clearMostRecentFocusOwnerOnHide,
containsFocus, countComponents, countHierarchyMembers, createChildHierarchyEvents,
createHierarchyEvents, decreaseComponentCount, deliverEvent, dispatchEventImpl,
dispatchEventToSelf, doLayout, eventEnabled, findComponentAt, findComponentAt,
findComponentAt, findComponentAtImpl, findTraversalRoot, getAccessibleAt,
getAccessibleChild, getAccessibleChildrenCount, getAlignmentX, getAlignmentY,
getBottommostComponentIndex, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents_NoClientCode, getComponents, getComponentZOrder,
getContainerListeners, getDropTargetEventTarget, getFocusTraversalKeys,
getFocusTraversalPolicy, getHeavyweightContainer, getInsets, getLayout, getListeners,
getListenersCount, getMaximumSize, getMinimumSize, getMouseEventTarget,
getMouseEventTarget, getMouseEventTargetImpl, getMousePosition, getOpaqueShape,
getPreferredSize, getTopmostComponentIndex, hasHeavyweightDescendants,
hasLightweightDescendants, increaseComponentCount, initializeFocusTraversalKeys,
initIDs, insets, invalidate, invalidateTree, isAncestorOf, isFocusCycleRoot,
isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet,
isParentOf, isRemoveNotifyNeeded, isSameOrAncestorOf, layout, lightweightPaint,
lightweightPrint, list, list, locate, minimumSize, mixOnShowing, mixOnValidating,
mixOnZOrderChanging, nextFocusHelper, numListening, paint, paintComponents,
paintHeavyweightComponents, paramString, postProcessKeyEvent, postsOldMouseEvents,
preferredSize, preProcessKeyEvent, print, printComponents,
printHeavyweightComponents, processContainerEvent, processEvent, proxyEnableEvents,
readObject, recursiveApplyCurrentShape, recursiveApplyCurrentShape,
recursiveApplyCurrentShape, recursiveSubtractAndApplyShape,
recursiveSubtractAndApplyShape, recursiveSubtractAndApplyShape, remove, remove,
removeAll, removeContainerListener, removeDelicately, removeNotify, reparentChild,
reparentTraverse, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout,
startLWModal, stopLWModal, transferFocusBackward, transferFocusDownCycle, update,
validate, validateTree, writeObject

```

**Methods inherited from class** java.awt.Component



removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, repaintParentIfNeeded, requestFocus, requestFocus, requestFocus, requestFocus, requestFocusHelper, requestFocusHelper, requestFocusInWindow, requestFocusInWindow, requestFocusInWindow, requestFocusInWindow, resetGC, reshape, reshapeNativePeer, resize, resize, setBackground, setBounds, setBounds, setBoundsOp, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys\_NoIDCheck, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setGCFFromPeer, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setRequestFocusController, setSize, setSize, setVisible, show, show, size, subtractAndApplyShape, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, updateCursorImmediately, validate, writeObject

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.awt.image.ImageObserver

imageUpdate

#### Methods inherited from interface java.awt.MenuContainer

getFont, postEvent, remove

#### Methods inherited from interface javax.accessibility.Accessible

getAccessibleContext

#### Methods inherited from interface java.awt.MenuContainer

getFont, postEvent, remove

#### Methods inherited from interface javax.accessibility.Accessible

getAccessibleContext

#### Methods inherited from interface javax.swing.RootPaneContainer

getContentPane, getGlassPane, getLayeredPane, getRootPane, setContentPane, setGlassPane, setLayeredPane

#### Methods inherited from interface javax.swing.TransferHandler.HasGetTransferHandler

getTransferHandler

#### Methods inherited from interface java.util.Observer

update

## Fields

## S

```
private ppstop_simulador.Simulador S
```

Referència al Simulador

---

## M

```
private ppstop_simulador.Model M
```

Referència al model de dades

---

## num\_linies

```
private int num_linies
```

Nombre de línies de la finestra de LOG

---

## jButton1

```
private javax.swing.JButton jButton1
```

---

## jLabel10

```
private javax.swing.JLabel jLabel10
```

---

## jLabel11

```
private javax.swing.JLabel jLabel11
```

---

## jLabel12

```
private javax.swing.JLabel jLabel12
```

---

## jLabel13

```
private javax.swing.JLabel jLabel13
```

---

## jLabel2

```
private javax.swing.JLabel jLabel2
```

---

## jLabel3

```
private javax.swing.JLabel jLabel3
```

---

---

## **jLabel4**

```
private javax.swing.JLabel jLabel4
```

---

---

## **jLabel5**

```
private javax.swing.JLabel jLabel5
```

---

---

## **jLabel6**

```
private javax.swing.JLabel jLabel6
```

---

---

## **jLabel7**

```
private javax.swing.JLabel jLabel7
```

---

---

## **jLabel8**

```
private javax.swing.JLabel jLabel8
```

---

---

## **jLabel9**

```
private javax.swing.JLabel jLabel9
```

---

---

## **jProgressBar1**

```
private javax.swing.JProgressBar jProgressBar1
```

---

---

## **jProgressBar2**

```
private javax.swing.JProgressBar jProgressBar2
```

---

---

## **jScrollPane2**

```
private javax.swing.JScrollPane jScrollPane2
```

---

## **jScrollPane3**

```
private javax.swing.JScrollPane jScrollPane3
```

---

## **JTable1**

```
private javax.swing.JTable JTable1
```

---

## **TextField2**

```
private javax.swing.JTextField TextField2
```

---

## **TextField3**

```
private javax.swing.JTextField TextField3
```

---

## **TextField4**

```
private javax.swing.JTextField TextField4
```

---

## **TextField5**

```
private javax.swing.JTextField TextField5
```

---

## **TextField6**

```
private javax.swing.JTextField TextField6
```

---

## **TextField7**

```
private javax.swing.JTextField TextField7
```

---

## **TextField8**

```
private javax.swing.JTextField TextField8
```

---

## **TextField9**

```
private javax.swing.JTextField TextField9
```

---

---

## jTextPanel

```
private javax.swing.JTextPane jTextPanel
```

## Constructors

### GUI

```
public GUI(Simulador S,  
           Model M)
```

Crea una nova instància de GUI

**Parameters:**

S - Referència a l'Simulador

M - Referència al Model de dades

## Methods

### afegirLog

```
public void afegirLog(java.lang.String text)
```

Afegeix una línia de text a la finestra de LOG

**Parameters:**

text - Línia de text a afegir

---

### update

```
public void update(java.util.Observable o,  
                   java.lang.Object arg)
```

Actualitza la vista amb les noves dades del Model (patró MVC)

**Parameters:**

o - Observable que notifica el canvi en el Model

arg - Arguments

---

### initComponents

```
private void initComponents()
```

Inicialitza els components de la GUI

---

### jButton1ActionPerformed

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
```

Acció d'activar el sensor d'humitat en polsar el botó de la GUI

**Parameters:**

evt - Event captat pel botó



---

### **jTextField7ActionPerformed**

```
private void jTextField7ActionPerformed(java.awt.event.ActionEvent evt)
```

---

### **jTextField8ActionPerformed**

```
private void jTextField8ActionPerformed(java.awt.event.ActionEvent evt)
```

## ppstop\_simulador Class Model

```

java.lang.Object
  |
  +- java.util.Observable
        |
        +- ppstop_simulador.Model
  
```

```

public class Model
extends java.util.Observable
  
```

Classe que conté tots els paràmetres del simulador. Actua com a Model dins del patró MVC (Model-View-Controller)

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juliol de 2009)

### Field Summary

private	<a href="#"><u>connexions_establertes</u></a> Nombre de connexions establertes pel dispositiu PPStop
private	<a href="#"><u>Dispositius</u></a>
private	<a href="#"><u>estat_bateria</u></a> Estat de la bateria del dispositiu PPStop (%)
private	<a href="#"><u>flag_inicial</u></a> Flag d'inici del dispositiu PPStop
private	<a href="#"><u>intents_connexio</u></a> Nombre d'intents de connexió del dispositiu PPStop
private	<a href="#"><u>MACSimulador</u></a> Adreça bluetooth del dispositiu PPStop
private	<a href="#"><u>num_dispositius_ppstop</u></a> Nombre de terminals PPStop associats al dispositiu PPStop
private	<a href="#"><u>peticions_busqueda</u></a> Nombre de peticions de buscar terminals rebudes pel dispositiu PPStop
private	<a href="#"><u>peticions_canviordre</u></a> Nombre de peticions de canvi d'ordre dels terminals associats
private	<a href="#"><u>peticions_sensibilitat</u></a> Nombre de peticions de canvi de sensibilitat rebudes pel dispositiu PPStop
private	<a href="#"><u>sensibilitat_sensor</u></a> Nivell de sensibilitat del sensor d'humitat del dispositiu PPStop (1-5)
private	<a href="#"><u>SensorButton</u></a> Indica si el botó d'activació del sensor d'humitat està activat

private	<a href="#">status</a> Status del dispositiu PPStop
---------	--

#### Fields inherited from class `java.util.Observable`

changed, obs

## Constructor Summary

public	<a href="#">Model()</a>
--------	-------------------------

## Method Summary

void	<a href="#">decrementEstatBateria(int num)</a> Decrementa l'estat de la bateria
void	<a href="#">enableSensorButton(boolean b)</a> Canvia l'estat del botó d'activació del sensor d'humitat
java.lang.Integer	<a href="#">getConnexionsEstablertes()</a> Obté el nombre de connexions establertes
java.util.Vector	<a href="#">getDispositiusPPStopTrobats()</a> Obté el vector de terminals PPStop trobats
java.lang.Integer	<a href="#">getEstatBateria()</a> Obté l'estat de la bateria
java.lang.String	<a href="#">getEstatBateriaTXT()</a> Obté l'estat de la bateria en format TXT de 3 dígit per a TX
java.lang.Integer	<a href="#">getFlagInicial()</a> Obté el flag inicial del dispositiu PPStop
java.lang.Integer	<a href="#">getIntentsConnexio()</a> Obté els nombre d'intents de connexió
java.lang.String	<a href="#">getMACSimulador()</a> Obté l'adreça bluetooth del dispositiu PPStop
int	<a href="#">getNumDispositiusPPStopTrobats()</a> Obté el nombre de terminals PPStop trobats
java.lang.Integer	<a href="#">getPetitionsBusqueda()</a> Obté el nombre de peticions de búsqueda de terminals PPStop
java.lang.Integer	<a href="#">getPetitionsCanviOrdre()</a> Obté el nombre de peticions de canvi d'ordre dels terminals PPStop associats
java.lang.Integer	<a href="#">getPetitionsSensibilitat()</a> Obté el nombre de peticions de canvi de sensibilitat
java.lang.Integer	<a href="#">getSensibilitatSensor()</a> Obté el nivell de sensibilitat del sensor d'humitat
java.lang.String	<a href="#">getSensibilitatSensorTXT()</a> Obté el nivell de sensibilitat del sensor d'humitat en format TXT per a TX

boolean	<a href="#"><u>getSensorButton()</u></a> Obté l'estat del botó d'activació del sensor d'humitat
java.lang.String	<a href="#"><u>getStatus()</u></a> Obté l'estatus del dispositiu PPStop
void	<a href="#"><u>incrementConnexionsEstablertes()</u></a> Incrementa el nombre de connexions establertes
void	<a href="#"><u>incrementIntentsConnexio()</u></a> Incrementa el nombre d'intents de connexió
void	<a href="#"><u>incrementNumDispositiusPPStopTrobats()</u></a> Incrementa el nombre de terminals PPStop trobats
void	<a href="#"><u>incrementPeticionsBusqueda()</u></a> Incrementa el nombre de peticions de búsqueda de terminals
void	<a href="#"><u>incrementPeticionsCanviOrdre()</u></a> Incrementa el nombre de peticions de canvi d'ordre dels terminals PPStop associats
void	<a href="#"><u>incrementPeticionsSensibilitat()</u></a> Incrementa el nombre de peticions de canvi de sensibilitat
void	<a href="#"><u>setDispositiusPPStopTrobats(java.util.Vector V)</u></a> Estableix el vector de terminals PPStop trobats
void	<a href="#"><u>setFlagInicial(java.lang.Integer flag)</u></a> Modifica el flag inicial del dispositiu PPStop
void	<a href="#"><u>setMACSimulador(java.lang.String Addr)</u></a> Defineix l'adreça bluetooth del dispositiu PPStop
void	<a href="#"><u>setNumDispositiusPPStopTrobats(int num)</u></a> Estableix el nombre de terminals PPStop trobats
void	<a href="#"><u>setSensibilitatSensor(int num)</u></a> Modifica el nivell de sensibilitat del sensor d'humitat
void	<a href="#"><u>setStatus(java.lang.String text)</u></a> Canvia l'estatus del dispositiu PPStop

#### Methods inherited from class java.util.Observable

addObserver, clearChanged, countObservers, deleteObserver, deleteObservers, hasChanged, notifyObservers, notifyObservers, setChanged

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

(continued from last page)

---

## flag\_inicial

```
private int flag_inicial
```

Flag d'inici del dispositiu PPStop

---

## num\_dispositius\_ppstop

```
private int num_dispositius_ppstop
```

Nombre de terminals PPStop associats al dispositiu PPStop

---

## intents\_connexio

```
private int intents_connexio
```

Nombre d'intents de connexió del dispositiu PPStop

---

## connexions\_establertes

```
private int connexions_establertes
```

Nombre de connexions establertes pel dispositiu PPStop

---

## peticions\_sensibilitat

```
private int peticions_sensibilitat
```

Nombre de peticions de canvi de sensibilitat rebudes pel dispositiu PPStop

---

## peticions\_busqueda

```
private int peticions_busqueda
```

Nombre de peticions de buscar terminals rebudes pel dispositiu PPStop

---

## peticions\_canviordre

```
private int peticions_canviordre
```

Nombre de peticions de canvi d'ordre dels terminals associats

---

## sensibilitat\_sensor

```
private int sensibilitat_sensor
```

Nivell de sensibilitat del sensor d'humitat del dispositiu PPStop (1-5)

---

## estat\_bateria

```
private int estat_bateria
```

Estat de la bateria del dispositiu PPStop (%)

---

## MACSimulador

```
private java.lang.String MACSimulador
```

---

(continued from last page)

Adreça bluetooth del dispositiu PPStop

---

## status

```
private java.lang.String status
```

Status del dispositiu PPStop

---

## SensorButton

```
private boolean SensorButton
```

Indica si el botó d'activació del sensor d'humitat està activat

---

## Dispositius

```
private java.util.Vector Dispositius
```

## Constructors

### Model

```
public Model()
```

## Methods

### enableSensorButton

```
public void enableSensorButton(boolean b)
```

Canvia l'estat del botó d'activació del sensor d'humitat

**Parameters:**

b - Estat del botó (booleà)

---

### getSensorButton

```
public boolean getSensorButton()
```

Obté l'estat del botó d'activació del sensor d'humitat

**Returns:**

Estat del botó d'activació (booleà)

---

### setStatus

```
public void setStatus(java.lang.String text)
```

Canvia l'estatus del dispositiu PPStop

**Parameters:**

text - Text que defineix l'estatus

---

---

## getStatus

```
public java.lang.String getStatus()
```

Obté l'estatus del dispositiu PPStop

**Returns:**

Text que defineix l'estatus

---

## setMACSimulador

```
public void setMACSimulador(java.lang.String Addr)
```

Defineix l'adreça bluetooth del dispositiu PPStop

**Parameters:**

Addr - Adreça bluetooth

---

## getMACSimulador

```
public java.lang.String getMACSimulador()
```

Obté l'adreça bluetooth del dispositiu PPStop

**Returns:**

Adreça bluetooth

---

## setFlagInicial

```
public void setFlagInicial(java.lang.Integer flag)
```

Modifica el flag inicial del dispositiu PPStop

**Parameters:**

flag - Valor del flag

---

## getFlagInicial

```
public java.lang.Integer getFlagInicial()
```

Obté el flag inicial del dispositiu PPStop

**Returns:**

Valor del flag

---

## incrementIntentsConnexio

```
public void incrementIntentsConnexio()
```

Incrementa el nombre d'intents de connexió

---

## getIntentsConnexio

```
public java.lang.Integer getIntentsConnexio()
```

Obté els nombre d'intents de connexió

---

(continued from last page)

**Returns:**

Nombre d'intents de connexió

---

**incrementConnexionsEstablertes**

```
public void incrementConnexionsEstablertes()
```

Incrementa el nombre de connexions establertes

---

**getConnexionsEstablertes**

```
public java.lang.Integer getConnexionsEstablertes()
```

Obté el nombre de connexions establertes

**Returns:**

Nombre de connexions establertes

---

**setSensibilitatSensor**

```
public void setSensibilitatSensor(int num)
```

Modifica el nivell de sensibilitat del sensor d'humitat

**Parameters:**

num - Nivell de sensibilitat (1-5)

---

**getSensibilitatSensor**

```
public java.lang.Integer getSensibilitatSensor()
```

Obté el nivell de sensibilitat del sensor d'humitat

**Returns:**

Nivell de sensibilitat (1-5)

---

**getSensibilitatSensorTXT**

```
public java.lang.String getSensibilitatSensorTXT()
```

Obté el nivell de sensibilitat del sensor d'humitat en format TXT per a TX

**Returns:**

Nivell de sensibilitat (1-5)

---

**incrementPeticionsSensibilitat**

```
public void incrementPeticionsSensibilitat()
```

Incrementa el nombre de peticions de canvi de sensibilitat

---

**getPeticionsSensibilitat**

```
public java.lang.Integer getPeticionsSensibilitat()
```

Obté el nombre de peticions de canvi de sensibilitat

**Returns:**



(continued from last page)

Nombre de peticions

---

## incrementPeticionsBusqueda

```
public void incrementPeticionsBusqueda()
```

Incrementa el nombre de peticions de búsqueda de terminals

---

## getPeticionsBusqueda

```
public java.lang.Integer getPeticionsBusqueda()
```

Obté el nombre de peticions de búsqueda de terminals PPStop

**Returns:**

Nombre de peticions

---

## incrementPeticionsCanviOrdre

```
public void incrementPeticionsCanviOrdre()
```

Incrementa el nombre de peticions de canvi d'ordre dels terminals PPStop associats

---

## getPeticionsCanviOrdre

```
public java.lang.Integer getPeticionsCanviOrdre()
```

Obté el nombre de peticions de canvi d'ordre dels terminals PPStop associats

**Returns:**

Nombre de peticions

---

## decrementEstatBateria

```
public void decrementEstatBateria(int num)
```

Decrementa l'estat de la bateria

**Parameters:**

num - Decrement de bateria (%)

---

## getEstatBateria

```
public java.lang.Integer getEstatBateria()
```

Obté l'estat de la bateria

**Returns:**

Estat de la bateria (%)

---

## getEstatBateriaTXT

```
public java.lang.String getEstatBateriaTXT()
```

Obté l'estat de la bateria en format TXT de 3 dígits per a TX

**Returns:**

Estat de la bateria (3 dígits)

---

## incrementNumDispositiusPPStopTrobat

```
public void incrementNumDispositiusPPStopTrobat()
```

Incrementa el nombre de terminals PPStop trobat

---

## setNumDispositiusPPStopTrobat

```
public void setNumDispositiusPPStopTrobat(int num)
```

Estableix el nombre de terminals PPStop trobat

**Parameters:**

num - Nombre de terminals

---

## getNumDispositiusPPStopTrobat

```
public int getNumDispositiusPPStopTrobat()
```

Obté el nombre de terminals PPStop trobat

**Returns:**

Nombre de terminals

---

## setDispositiusPPStopTrobat

```
public void setDispositiusPPStopTrobat(java.util.Vector v)
```

Estableix el vector de terminals PPStop trobat

**Parameters:**

v - Vector de terminals PPStop

---

## getDispositiusPPStopTrobat

```
public java.util.Vector getDispositiusPPStopTrobat()
```

Obté el vector de terminals PPStop trobat

**Returns:**

Vector de terminals PPStop

---

# ppstop\_simulador

## Class Simulador

java.lang.Object

└─ppstop\_simulador.Simulador

public class **Simulador**  
extends java.lang.Object

Classe principal del simulador

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juliol de 2009)

### Field Summary

private static	<a href="#">CS</a> Objecte per a la Comunicació Sèrie amb el transmissor BT
private static	<a href="#">M</a> Model de dades (patró MVC)
private static	<a href="#">UC</a> Unitat de Control: lògica del dispositiu PPStop simulat
private static	<a href="#">V</a> GUI del simulador (patró MVC)

### Constructor Summary

public	<a href="#">Simulador()</a> Crea una instància de Simulador
--------	--

### Method Summary

void	<a href="#">activarSensor()</a> Actua segons l'estat del protocol
void	<a href="#">enviarAT(java.lang.String comandaAT)</a> Envia una comanda AT al transmissor BT
void	<a href="#">finalitzar()</a> Tanca el port sèrie i surt del simulador
void	<a href="#">iniciar()</a> Obte el port sèrie i inicia el protocol
void	<a href="#">log(java.lang.String s)</a> Afegeix una línia de text al LOG de pantalla
static void	<a href="#">main(java.lang.String[] args)</a> Programa principal del simulador

void

[setResposta](#)(java.lang.String resposta)

Comunica a la Unitat de Control la resposta obtinguda pel transmissor BT

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### CS

private static ppstop\_simulador.ComunicacioSerie **CS**

Objecte per a la Comunicació Sèrie amb el transmissor BT

### UC

private static ppstop\_simulador.UnitatControl **UC**

Unitat de Control: lògica del dispositiu PPStop simulat

### M

private static ppstop\_simulador.Model **M**

Model de dades (patró MVC)

### V

private static ppstop\_simulador.GUI **V**

GUI del simulador (patró MVC)

## Constructors

### Simulador

public **Simulador**()

Crea una instància de Simulador

## Methods

### iniciar

public void **iniciar**()

Obte el port sèrie i inicia el protocol

### finalitzar

public void **finalitzar**()

Tanca el port sèrie i surt del simulador

---

## enviarAT

```
public void enviarAT(java.lang.String comandaAT)
```

Envia una comanda AT al transmissor BT

### Parameters:

comandaAT - Comanda AT a enviar

---

## setResposta

```
public void setResposta(java.lang.String resposta)
```

Comunica a la Unitat de Control la resposta obtinguda pel transmissor BT

### Parameters:

resposta - Resposta obtinguda

---

## activarSensor

```
public void activarSensor()
```

Actua segons l'estat del protocol

---

## log

```
public void log(java.lang.String s)
```

Afegeix una línia de text al LOG de pantalla

### Parameters:

s - Línia de text a afegir al LOG

---

## main

```
public static void main(java.lang.String[] args)
```

Programa principal del simulador

### Parameters:

args - Arguments del simulador

---

# ppstop\_simulador

## Class UnitatControl

java.lang.Object

└─ppstop\_simulador.UnitatControl

```
public class UnitatControl
extends java.lang.Object
```

Classe que conté la Unitat de Control del dispositiu PPStop simulat

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juliol de 2009)

### Field Summary

private static final	<a href="#">COMANDA_BUSQUEDASERVEIS_1</a> Comanda AT (parcial 1) d'inici de búsqueda de serveis en un determinat terminal Value: <b>AT+JSDS=</b>
private static final	<a href="#">COMANDA_BUSQUEDASERVEIS_2</a> Comanda AT (parcial 2) d'inici de búsqueda de serveis en un determinat terminal Value: <b>,1101</b>
private static final	<a href="#">COMANDA_BUSQUEDATERMINALS</a> Comanda AT d'inici de búsqueda de terminals Value: <b>AT+JDDS=0</b>
private static final	<a href="#">COMANDA_CONNEXIO</a> Comanda AT (parcial) per a iniciar una connexió bluetooth Value: <b>AT+JCCR=</b>
private static final	<a href="#">COMANDA_DATA_ALERTAR</a> Comanda AT (parcial) per a enviar l'ordre d'alarmar Value: <b>AT+JSDA=005,A</b>
private static final	<a href="#">COMANDA_DATA_COMPROVAR</a> Comanda AT (parcial) per a enviar l'ordre de comprovar si un terminal PPStop és un terminal autoritzat Value: <b>AT+JSDA=005,C</b>
private static final	<a href="#">COMANDA_GETADRESS</a> Comanda AT d'obtenció de l'adreça bluetooth Value: <b>AT+JRBD</b>
private static final	<a href="#">COMANDA_INIPINS</a> Comanda AT d'inicialització dels pins Value: <b>AT+JGPC=0000,0000,0000,0000,0000</b>
private static final	<a href="#">COMANDA_POSARNOM</a> Comanda AT d'establiment del nom del dispositiu Value: <b>AT+JSLN=10,PPSTOP_XIP</b>

private static final	<a href="#"><u>COMANDA_RESET</u></a> Comanda AT de reset Value: <b>AT+JRES</b>
private static final	<a href="#"><u>COMANDA_SEGURETAT</u></a> Comanda AT d'establiment de seguretat: sense autenticació ni encriptació Value: <b>AT+JSEC=1,1,1,04,1111</b>
private	<a href="#"><u>dispositiu_PPStop</u></a> Indica si el terminal que s'està analitzant és un terminal PPStop
private	<a href="#"><u>dispositiu_PPStop_autoritzat</u></a> Indica si el terminal que s'està analitzant és un terminal PPStop autoritzat
private	<a href="#"><u>Dispositius</u></a> Vector de terminals PPStop
private	<a href="#"><u>estat</u></a> Estat actual de la màquina d'estats
private static final	<a href="#"><u>ESTAT_BUSQUEDASERVEIS</u></a> Estat de búsqueda de servei PPStop en els terminals trobats Value: <b>7</b>
private static final	<a href="#"><u>ESTAT_BUSQUEDASERVEIS_RESP</u></a> Estat d'espera de resposta a la búsqueda de servei PPStop en els terminals trobats Value: <b>8</b>
private static final	<a href="#"><u>ESTAT_BUSQUEDATERMINALS</u></a> Estat de búsqueda de terminals en àrea de cobertura Value: <b>5</b>
private static final	<a href="#"><u>ESTAT_BUSQUEDATERMINALS_RESP</u></a> Estat d'espera de resposta a la búsqueda de terminals en àrea de cobertura Value: <b>6</b>
private static final	<a href="#"><u>ESTAT_CANVIARSENSIBILITAT</u></a> Estat de canviar la sensibilitat del sensor d'humitat del dispositiu PPStop Value: <b>22</b>
private static final	<a href="#"><u>ESTAT_COMPROVARSERVEI</u></a> Estat de comprovació de servei PPStop autoritzat Value: <b>9</b>
private static final	<a href="#"><u>ESTAT_COMPROVARSERVEI_RESP1</u></a> Estat d'espera de resposta (1) a la comprovació de servei PPStop autoritzat Value: <b>10</b>
private static final	<a href="#"><u>ESTAT_COMPROVARSERVEI_RESP2</u></a> Estat d'espera de resposta (2) a la comprovació de servei PPStop autoritzat Value: <b>11</b>
private static final	<a href="#"><u>ESTAT_DEFINIRSEGURETAT</u></a> Estat de definició de la seguretat de les comunicacions bluetooth Value: <b>2</b>

private static final	<a href="#"><u>ESTAT_ESPERAREVENT</u></a> Estat d'espera d'event (activar el sensor d'humitat) Value: <b>15</b>
private static final	<a href="#"><u>ESTAT_ESTABLIRCONNEXIO</u></a> Estat d'establir una connexió d'alarma als terminals PPStop associats Value: <b>30</b>
private static final	<a href="#"><u>ESTAT_ESTABLIRCONNEXIO_RESP1</u></a> Estat d'espera de resposta (1) a la connexió d'alarma als terminals PPStop associats Value: <b>31</b>
private static final	<a href="#"><u>ESTAT_ESTABLIRCONNEXIO_RESP2</u></a> Estat d'espera de resposta (2) a la connexió d'alarma als terminals PPStop associats Value: <b>32</b>
private static final	<a href="#"><u>ESTAT_GETADRESS</u></a> Estat d'obtenció de l'adreça bluetooth del dispositiu (transmissor bluetooth) Value: <b>0</b>
private static final	<a href="#"><u>ESTAT_INIPINS</u></a> Estat d'inicialitzar els pins del transmissor Value: <b>3</b>
private static final	<a href="#"><u>ESTAT_POSARNOM</u></a> Estat de definir el nom del dispositiu bluetooth Value: <b>4</b>
private static final	<a href="#"><u>ESTAT_RESET</u></a> Estat de reset del dispositiu PPStop (transmissor bluetooth) Value: <b>1</b>
private	<a href="#"><u>estatFutur</u></a> Estat futur de la màquina d'estats
private	<a href="#"><u>i</u></a> Variable auxiliar per als bucles
private	<a href="#"><u>M</u></a> Referència al Model de dades
private	<a href="#"><u>resposta</u></a> Resposta rebuda del transmissor bluetooth
private	<a href="#"><u>S</u></a> Referència al Simulador
private	<a href="#"><u>sensibilitatRebuda</u></a> Nivell de sensibilitat rebut
private	<a href="#"><u>tfinal_connexio</u></a> Variable per al control del temps total de connexió
private	<a href="#"><u>tinicial_connexio</u></a> Variable per al control del temps total de connexió

## Constructor Summary



public	<a href="#">UnitatControl</a> ( <a href="#">Simulador</a> S, <a href="#">Model</a> M) Crea una instància de UnitatControl
--------	--

## Method Summary

void	<a href="#">activarSensor()</a> Activa el sensor d'humitat actuant consegüentment segons l'estat del protocol
void	<a href="#">actuar()</a> Màquina d'estats: canvia d'estat segons la resposta obtinguda i actua en conseqüència
void	<a href="#">setResposta</a> (java.lang.String resposta) Estableix la resposta obtinguda pel port sèrie

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### S

private ppstop\_simulador.Simulador **S**  
Referència al Simulador

### M

private ppstop\_simulador.Model **M**  
Referència al Model de dades

### estat

private int **estat**  
Estat actual de la màquina d'estats

### estatFutur

private int **estatFutur**  
Estat futur de la màquina d'estats

### sensibilitatRebuda

private int **sensibilitatRebuda**  
Nivell de sensibilitat rebut

### resposta

private java.lang.String **resposta**

(continued from last page)

Resposta rebuda del transmissor bluetooth

---

## Dispositius

```
private java.util.Vector Dispositius
```

Vector de terminals PPStop

---

## dispositiu\_PPStop

```
private boolean dispositiu_PPStop
```

Indica si el terminal que s'està analitzant és un terminal PPStop

---

## dispositiu\_PPStop\_autoritzat

```
private boolean dispositiu_PPStop_autoritzat
```

Indica si el terminal que s'està analitzant és un terminal PPStop autoritzat

---

## tinicial\_connexio

```
private long tinicial_connexio
```

Variable per al control del temps total de connexió

---

## tfinal\_connexio

```
private long tfinal_connexio
```

Variable per al control del temps total de connexió

---

## i

```
private int i
```

Variable auxiliar per als bucles

---

## ESTAT\_GETADDRESS

```
private static final int ESTAT_GETADDRESS
```

Estat d'obtenció de l'adreça bluetooth del dispositiu (transmissor bluetooth)  
Constant value: 0

---

## ESTAT\_RESET

```
private static final int ESTAT_RESET
```

Estat de reset del dispositiu PPStop (transmissor bluetooth)  
Constant value: 1

---

## ESTAT\_DEFINIRSEGURETAT

```
private static final int ESTAT_DEFINIRSEGURETAT
```

Estat de definició de la seguretat de les comunicacions bluetooth  
Constant value: 2

---

## ESTAT\_INIPINS

private static final int **ESTAT\_INIPINS**

Estat d'inicialitzar els pins del transmissor  
Constant value: **3**

---

## ESTAT\_POSARNOM

private static final int **ESTAT\_POSARNOM**

Estat de definir el nom del dispositiu bluetooth  
Constant value: **4**

---

## ESTAT\_BUSQUEDATERMINALS

private static final int **ESTAT\_BUSQUEDATERMINALS**

Estat de búsqueda de terminals en àrea de cobertura  
Constant value: **5**

---

## ESTAT\_BUSQUEDATERMINALS\_RESP

private static final int **ESTAT\_BUSQUEDATERMINALS\_RESP**

Estat d'espera de resposta a la búsqueda de terminals en àrea de cobertura  
Constant value: **6**

---

## ESTAT\_BUSQUEDASERVEIS

private static final int **ESTAT\_BUSQUEDASERVEIS**

Estat de búsqueda de servei PPStop en els terminals trobats  
Constant value: **7**

---

## ESTAT\_BUSQUEDASERVEIS\_RESP

private static final int **ESTAT\_BUSQUEDASERVEIS\_RESP**

Estat d'espera de resposta a la búsqueda de servei PPStop en els terminals trobats  
Constant value: **8**

---

## ESTAT\_COMPROVARSERVEI

private static final int **ESTAT\_COMPROVARSERVEI**

Estat de comprovació de servei PPStop autoritzat  
Constant value: **9**

---

## ESTAT\_COMPROVARSERVEI\_RESP1

private static final int **ESTAT\_COMPROVARSERVEI\_RESP1**

Estat d'espera de resposta (1) a la comprovació de servei PPStop autoritzat  
Constant value: **10**

---

(continued from last page)

---

## ESTAT\_COMPROVARSERVEI\_RESP2

```
private static final int ESTAT_COMPROVARSERVEI_RESP2
```

Estat d'espera de resposta (2) a la comprovació de servei PPStop autoritzat  
Constant value: **11**

---

## ESTAT\_ESPERAREVENT

```
private static final int ESTAT_ESPERAREVENT
```

Estat d'espera d'event (activar el sensor d'humitat)  
Constant value: **15**

---

## ESTAT\_CANVIARSENSIBILITAT

```
private static final int ESTAT_CANVIARSENSIBILITAT
```

Estat de canviar la sensibilitat del sensor d'humitat del dispositiu PPStop  
Constant value: **22**

---

## ESTAT\_ESTABLIRCONNEXIO

```
private static final int ESTAT_ESTABLIRCONNEXIO
```

Estat d'establir una connexió d'alarma als terminals PPStop associats  
Constant value: **30**

---

## ESTAT\_ESTABLIRCONNEXIO\_RESP1

```
private static final int ESTAT_ESTABLIRCONNEXIO_RESP1
```

Estat d'espera de resposta (1) a la connexió d'alarma als terminals PPStop associats  
Constant value: **31**

---

## ESTAT\_ESTABLIRCONNEXIO\_RESP2

```
private static final int ESTAT_ESTABLIRCONNEXIO_RESP2
```

Estat d'espera de resposta (2) a la connexió d'alarma als terminals PPStop associats  
Constant value: **32**

---

## COMANDA\_GETADRESS

```
private static final java.lang.String COMANDA_GETADRESS
```

Comanda AT d'obtenció de l'adreça bluetooth  
Constant value: **AT+JRBD**

---

## COMANDA\_RESET

```
private static final java.lang.String COMANDA_RESET
```

Comanda AT de reset  
Constant value: **AT+JRES**

---

## COMANDA\_INIPINS

```
private static final java.lang.String COMANDA_INIPINS
```

---

(continued from last page)

Comanda AT d'inicialització dels pins  
Constant value: **AT+JGPC=0000,0000,0000,0000,0000**

---

## COMANDA\_SEGURETAT

```
private static final java.lang.String COMANDA_SEGURETAT
```

Comanda AT d'establiment de seguretat: sense autenticació ni encriptació  
Constant value: **AT+JSEC=1,1,1,04,1111**

---

## COMANDA\_POSARNOM

```
private static final java.lang.String COMANDA_POSARNOM
```

Comanda AT d'establiment del nom del dispositiu  
Constant value: **AT+JSLN=10,PPSTOP\_XIP**

---

## COMANDA\_BUSQUEDATERMINALS

```
private static final java.lang.String COMANDA_BUSQUEDATERMINALS
```

Comanda AT d'inici de búsqueda de terminals  
Constant value: **AT+JDDS=0**

---

## COMANDA\_BUSQUEDASERVEIS\_1

```
private static final java.lang.String COMANDA_BUSQUEDASERVEIS_1
```

Comanda AT (parcial 1) d'inici de búsqueda de serveis en un determinat terminal  
Constant value: **AT+JSDS=**

---

## COMANDA\_BUSQUEDASERVEIS\_2

```
private static final java.lang.String COMANDA_BUSQUEDASERVEIS_2
```

Comanda AT (parcial 2) d'inici de búsqueda de serveis en un determinat terminal  
Constant value: **,1101**

---

## COMANDA\_DATA\_COMPROVAR

```
private static final java.lang.String COMANDA_DATA_COMPROVAR
```

Comanda AT (parcial) per a enviar l'ordre de comprovar si un terminal PPStop és un terminal autoritzat  
Constant value: **AT+JSDA=005,C**

---

## COMANDA\_DATA\_ALERTAR

```
private static final java.lang.String COMANDA_DATA_ALERTAR
```

Comanda AT (parcial) per a enviar l'ordre d'alarmar  
Constant value: **AT+JSDA=005,A**

---

## COMANDA\_CONNEXIO

```
private static final java.lang.String COMANDA_CONNEXIO
```

Comanda AT (parcial) per a iniciar una connexió bluetooth  
Constant value: **AT+JCCR=**

(continued from last page)

## Constructors

### UnitatControl

```
public UnitatControl(Simulador S,  
                    Model M)
```

Crea una instància de UnitatControl

**Parameters:**

S - Referència al Simulador

M - Referència al Model de dades

## Methods

### setResposta

```
public void setResposta(java.lang.String resposta)
```

Estableix la resposta obtinguda pel port sèrie

**Parameters:**

resposta - Resposta obtinguda

---

### activarSensor

```
public void activarSensor()
```

Activa el sensor d'humitat actuant consegüentment segons l'estat del protocol

---

### actuar

```
public void actuar()
```

Màquina d'estats: canvia d'estat segons la resposta obtinguda i actua en conseqüència

## ppstop\_simulador

### Interface Vista

All Known Implementing Classes:

[GUI](#)

---

public interface **Vista**  
extends java.util.Observer

Interfície d'un observador per a la Vista (GUI) del patró MVC

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juliol de 2009)

---

Methods inherited from interface <code>java.util.Observer</code>
update

---

## A.3 Codi del simulador

Codi A.1: (Simulador) ComunicacioSerie.java

```
1 package ppstop_simulador;
2
3 import java.io.*;
4 import java.util.*;
5 import java.util.logging.Level;
6 import java.util.logging.Logger;
7 import javax.comm.*;
8
9 /**
10  * Classe per a la Comunicació Sèrie amb el transmissor BT
11  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
12  * @version 1.0 (Juliol de 2009)
13  */
14 public class ComunicacioSerie implements SerialPortEventListener {
15
16     /** Referència al simulador */
17     private Simulador S;
18     /** Identificador del port sèrie */
19     private CommPortIdentifier idPort;
20     /** Objecte de control del Port Sèrie */
21     private SerialPort portSerie;
22     /** Bufer de recepció de dades */
23     private BufferedReader entrada;
24     /** Stream de sortida de dades */
25     private OutputStream sortida;
26
27     /**
28      * Crea una instància de ComunicacióSerie
29      * @param S Referència al simulador
30      */
31     public ComunicacioSerie(Simulador S){
32         this.S = S;
33     }
34
35     /**
36      * Obre una comunicació per un port sèrie determinat
37      * @param port Identificador del port a obrir
38      */
39     public void obrirPort(String port){
40         try {
41             idPort = CommPortIdentifier.getPortIdentifier(port);
42             portSerie = (SerialPort)idPort.open("PPStop",10000);
43             portSerie.setSerialPortParams(
44                 115200,SerialPort.DATABITS_8,
45                 SerialPort.STOPBITS_1,
46                 SerialPort.PARITY_NONE
47             );
48             entrada = new BufferedReader(new InputStreamReader(portSerie.getInputStream()));
49             sortida = portSerie.getOutputStream();
50             portSerie.addEventListener(this);
51             portSerie.notifyOnDataAvailable(true);
52             S.log("Port " + port + " obert");
53         } catch (TooManyListenersException ex) {
54             Logger.getLogger(Simulador.class.getName()).log(Level.SEVERE, null, ex);
55         } catch (IOException ex) {
56             Logger.getLogger(Simulador.class.getName()).log(Level.SEVERE, null, ex);
57         } catch (UnsupportedCommOperationException ex) {
58             Logger.getLogger(Simulador.class.getName()).log(Level.SEVERE, null, ex);
59         } catch (PortInUseException ex) {
60             Logger.getLogger(Simulador.class.getName()).log(Level.SEVERE, null, ex);
61         } catch (NoSuchPortException ex) {
62             Logger.getLogger(Simulador.class.getName()).log(Level.SEVERE, null, ex);
63         }
64     }
65 }
```



```

65
66  /**
67   * Tanca la comunicació pel port sèrie
68   */
69  public void tancarPort(){
70      portSerie.close();
71      S.log("Port tancat");
72  }
73
74  /**
75   * Envia una comanda AT al transmissor BT
76   * @param comanda      Comanda AT a enviar
77   */
78  public void enviarAT(String comanda){
79      try {
80          S.log("Enviat: " + comanda);
81          comanda += "\r\n";
82          sortida.write(comanda.getBytes());
83      } catch (IOException ex) {
84          Logger.getLogger(Simulador.class.getName()).log(Level.SEVERE, null, ex);
85      }
86  }
87
88  /**
89   * Llegeix la resposta del transmissor BT i ho notifica al simulador
90   */
91  public void llegirResposta(){
92      try {
93          String resposta;
94          do {
95              resposta = entrada.readLine();
96              S.log("Rebut: " + resposta);
97              S.setResposta(resposta);
98          } while (resposta!=null);
99      } catch (IOException ex) {
100          Logger.getLogger(Simulador.class.getName()).log(Level.SEVERE, null, ex);
101      }
102  }
103
104  /**
105   * Detecta un event en el Port Sèrie i actua en conseqüència
106   * @param event      Event detectat
107   */
108  public void serialEvent(SerialPortEvent event){
109      switch (event.getEventType()) {
110          case SerialPortEvent.DATA_AVAILABLE:
111              llegirResposta();
112              break;
113      }
114  }
115
116 }

```

## Codi A.2: (Simulador) Dispositiu.java

```

1 package ppstop_simulador;
2
3 /**
4  * Classe representativa d'un dispositiu bluetooth (terminal mòbil PPStop)
5  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
6  * @version 1.0 (Juliol de 2009)
7  */
8 public class Dispositiu {
9
10     /** Ordre del dispositiu */
11     private int DispOrd;
12     /** Nom del dispositiu */
13     private String DispNom;
14     /** Adreça bluetooth del dispositiu */

```

```

15 private String DispAddr;
16 /** Canal del servei PPStop del dispositiu */
17 private String DispChan;
18
19 /**
20  * Crea una instància de Dispositiu
21  * @param DispOrd      Ordre del dispositiu
22  * @param DispNom      Nom del dispositiu
23  * @param DispAddr     Adreça bluetooth del dispositiu
24  * @param DispChan     Canal del servei PPStop del dispositiu
25  */
26 public Dispositiu(int DispOrd, String DispNom, String DispAddr, String DispChan){
27     this.DispOrd = DispOrd;
28     this.DispNom = DispNom;
29     this.DispAddr = DispAddr;
30     this.DispChan = DispChan;
31 }
32
33 /**
34  * Estableix l'ordre del dispositiu
35  * @param n            Ordre del dispositiu
36  */
37 public void setDispOrd(int n){
38     DispOrd = n;
39 }
40
41 /**
42  * Estableix el canal del dispositiu
43  * @param ch           Canal del dispositiu
44  */
45 public void setDispChan(String ch){
46     DispChan = ch;
47 }
48
49 /**
50  * Obté l'ordre del dispositiu
51  * @return Ordre del dispositiu
52  */
53 public Integer getDispOrd(){
54     return DispOrd;
55 }
56
57 /**
58  * Obté el nom del dispositiu
59  * @return Nom del dispositiu
60  */
61 public String getDispNom(){
62     return DispNom;
63 }
64
65 /**
66  * Obté l'adreça bluetooth del dispositiu
67  * @return Adreça bluetooth del dispositiu
68  */
69 public String getDispAddr(){
70     return DispAddr;
71 }
72
73 /**
74  * Obté el canal del servei PPStop del dispositiu
75  * @return Número de canal del servei PPStop
76  */
77 public String getDispChan(){
78     return DispChan;
79 }
80
81 }

```

Codi A.3: (Simulador) GUI.java

```

1 package ppstop_simulador;
2
3 import java.util.Observable;
4 import javax.swing.ImageIcon;
5 import javax.swing.JFrame;
6 import javax.swing.table.AbstractTableModel;
7
8 /**
9  * Classe que genera la Graphic User Interface (GUI) de la Vista del simulador (patró
10  * MVC)
11  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
12  * @version 1.0 (Juliol de 2009)
13  */
14 public class GUI extends JFrame implements Vista {
15
16     /** Referència al Simulador */
17     private Simulador S;
18     /** Referència al model de dades */
19     private Model M;
20     /** Nombre de línies de la finestra de LOG */
21     private int num_linies = 0;
22
23     /**
24      * Crea una nova instància de GUI
25      * @param S Referència a l'Simulador
26      * @param M Referència al Model de dades
27      */
28     public GUI(Simulador S, Model M) {
29         this.S = S;
30         this.M = M;
31         this.setIconImage(new ImageIcon("icon.png").getImage());
32         initComponents();
33     }
34
35     /**
36      * Afegeix una línia de text a la finestra de LOG
37      * @param text Línia de text a afegir
38      */
39     public void afegirLog(String text){
40         JTextPanel1.setText(JTextPanel1.getText() + (++num_linies) + text + "\n");
41         JTextPanel1.setCaretPosition(JTextPanel1.getText().length());
42     }
43
44     /**
45      * Actualitza la vista amb les noves dades del Model (patró MVC)
46      * @param o Observable que notifica el canvi en el Model
47      * @param arg Arguments
48      */
49     public void update(Observable o, Object arg) {
50         jButton1.setEnabled(M.getSensorButton());
51         jLabel7.setText(M.getStatus());
52         jTextField2.setText(M.getIntentsConnexio().toString());
53         jTextField3.setText(M.getConnexionsEstablertes().toString());
54         jTextField4.setText(M.getPeticionsSensibilitat().toString());
55         jTextField5.setText(M.getPeticionsBusqueda().toString());
56         jTextField6.setText(M.getMACSimulador());
57         jTextField7.setText(M.getFlagInicial().toString());
58         jTextField8.setText(String.valueOf(M.getNumDispositiusPPStopTrobats()));
59         jTextField9.setText(M.getPeticionsCanviOrdre().toString());
60         jProgressBar1.setValue(M.getSensibilitatSensor()*10);
61         jProgressBar2.setValue(M.getEstatBateria());
62         jTable1.setModel(new AbstractTableModel() {
63             protected String[] columnNames = {"Ordre", "Dispositiu", "@MAC"};
64             @Override public String getColumnName(int col) { return columnNames[col].
65                 toString(); }
66             public int getRowCount() { return M.getDispositiusPPStopTrobats().size(); }
67             public int getColumnCount() { return columnNames.length; }
68             public Object getValueAt(int row, int col) {
69                 switch(col){

```

```

68         case 0:
69             return ( " " + M.getDispositiusPPStopTrobats().get(row).
getDispOrd());
70         case 1:
71             return ( " " + M.getDispositiusPPStopTrobats().get(row).
getDispNom());
72         case 2:
73             return ( " " + M.getDispositiusPPStopTrobats().get(row).
getDispAddr());
74         default:
75             return "--";
76     }
77 }
78 });
79 for (int i = 0; i < jTable1.getColumnCount(); i++) {
80     switch(i){
81         case 0:
82             jTable1.getColumnModel().getColumn(i).setPreferredWidth((20*jTable1.
getWidth())/100);
83             break;
84         case 1:
85             jTable1.getColumnModel().getColumn(i).setPreferredWidth((40*jTable1.
getWidth())/100);
86             break;
87         case 2:
88             jTable1.getColumnModel().getColumn(i).setPreferredWidth((40*jTable1.
getWidth())/100);
89             break;
90         default:
91             jTable1.getColumnModel().getColumn(i).setPreferredWidth((33*jTable1.
getWidth())/100);
92             break;
93     }
94 }
95 }
96
97
98 /**
99  * Inicialitza els components de la GUI
100  */
101 // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:
initComponents
102 private void initComponents() {
103
104     jButton1 = new javax.swing.JButton();
105     jLabel2 = new javax.swing.JLabel();
106     jLabel3 = new javax.swing.JLabel();
107     jLabel5 = new javax.swing.JLabel();
108     jTextField2 = new javax.swing.JTextField();
109     jTextField3 = new javax.swing.JTextField();
110     jTextField4 = new javax.swing.JTextField();
111     jTextField5 = new javax.swing.JTextField();
112     jLabel4 = new javax.swing.JLabel();
113     jLabel6 = new javax.swing.JLabel();
114     jLabel7 = new javax.swing.JLabel();
115     jLabel8 = new javax.swing.JLabel();
116     jTextField6 = new javax.swing.JTextField();
117     jScrollPane2 = new javax.swing.JScrollPane();
118     jPanel1 = new javax.swing.JPanel();
119     jLabel9 = new javax.swing.JLabel();
120     jLabel10 = new javax.swing.JLabel();
121     jTextField7 = new javax.swing.JTextField();
122     jTextField8 = new javax.swing.JTextField();
123     jScrollPane3 = new javax.swing.JScrollPane();
124     jTable1 = new javax.swing.JTable();
125     jLabel11 = new javax.swing.JLabel();
126     progressBar1 = new javax.swing.JProgressBar();
127     jLabel12 = new javax.swing.JLabel();
128     progressBar2 = new javax.swing.JProgressBar();
129     jLabel13 = new javax.swing.JLabel();

```

```

130     jTextField9 = new javax.swing.JTextField();
131
132     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
133     setTitle("Simulador PPStop");
134
135     jButton1.setText("Activar Sensor");
136     jButton1.setEnabled(false);
137     jButton1.addActionListener(new java.awt.event.ActionListener() {
138         public void actionPerformed(java.awt.event.ActionEvent evt) {
139             jButton1ActionPerformed(evt);
140         }
141     });
142
143     jLabel2.setText("Intents de Connexió");
144
145     jLabel3.setText("Connexions Establertes");
146
147     jLabel5.setText("Peticions Sensibilitat");
148
149     jTextField2.setEditable(false);
150     jTextField2.setText("0");
151
152     jTextField3.setEditable(false);
153     jTextField3.setText("0");
154
155     jTextField4.setEditable(false);
156     jTextField4.setText("0");
157
158     jTextField5.setEditable(false);
159     jTextField5.setText("0");
160
161     jLabel4.setText("Peticions Búsqueda");
162
163     jLabel6.setText("Dispositius PPStop trobats");
164
165     jLabel7.setForeground(new java.awt.Color(255, 0, 0));
166     jLabel7.setText("Status");
167
168     jLabel8.setText("@MAC Simulador");
169
170     jTextField6.setEditable(false);
171
172     jTextPane1.setEditable(false);
173     jTextPane1.setFont(new java.awt.Font("Tahoma", 1, 10));
174     jTextPane1.setForeground(new java.awt.Color(0, 51, 204));
175     jScrollPane2.setViewportView(jTextPane1);
176
177     jLabel9.setText("Flag Inicial");
178
179     jLabel10.setText("Dispositius Trobats");
180
181     jTextField7.setEditable(false);
182     jTextField7.setText("1");
183     jTextField7.addActionListener(new java.awt.event.ActionListener() {
184         public void actionPerformed(java.awt.event.ActionEvent evt) {
185             jTextField7ActionPerformed(evt);
186         }
187     });
188
189     jTextField8.setEditable(false);
190     jTextField8.setText("0");
191     jTextField8.addActionListener(new java.awt.event.ActionListener() {
192         public void actionPerformed(java.awt.event.ActionEvent evt) {
193             jTextField8ActionPerformed(evt);
194         }
195     });
196
197     jScrollPane3.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.
198         VERTICAL_SCROLLBAR_NEVER);

```

```

199 jTable1.setFont(new java.awt.Font("Tahoma", 0, 10));
200 jTable1.setModel(new javax.swing.table.DefaultTableModel(
201     new Object [][] {
202
203     },
204     new String [] {
205         "Ordre", "Dispositiu", "@MAC"
206     }
207 ) {
208     Class[] types = new Class [] {
209         java.lang.Integer.class, java.lang.String.class, java.lang.String.class
210     };
211     boolean[] canEdit = new boolean [] {
212         false, false, false
213     };
214
215     public Class getColumnClass(int columnIndex) {
216         return types [columnIndex];
217     }
218
219     public boolean isCellEditable(int rowIndex, int columnIndex) {
220         return canEdit [columnIndex];
221     }
222 });
223 jTable1.setEnabled(false);
224 jTable1.setGridColor(new java.awt.Color(204, 204, 204));
225 jTable1.getTableHeader().setResizingAllowed(false);
226 jTable1.getTableHeader().setReorderingAllowed(false);
227 jScrollPane3.setViewportViewView(jTable1);
228
229 jLabel11.setText("Sensibilitat Sensor");
230
231 progressBar1.setBackground(new java.awt.Color(204, 204, 204));
232 progressBar1.setFont(new java.awt.Font("Tahoma", 0, 10)); // NOI18N
233 progressBar1.setForeground(new java.awt.Color(255, 0, 51));
234 progressBar1.setValue(30);
235 progressBar1.setBorderPainted(false);
236 progressBar1.setStringPainted(true);
237
238 jLabel12.setText("Estat Bateria");
239
240 progressBar2.setBackground(new java.awt.Color(204, 204, 204));
241 progressBar2.setFont(new java.awt.Font("Tahoma", 0, 10));
242 progressBar2.setForeground(new java.awt.Color(0, 51, 204));
243 progressBar2.setValue(100);
244 progressBar2.setBorderPainted(false);
245 progressBar2.setStringPainted(true);
246
247 jLabel13.setText("Peticions Canvi d'Ordre");
248
249 jTextField9.setEditable(false);
250 jTextField9.setText("0");
251
252 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
253 getContentPane().setLayout(layout);
254 layout.setHorizontalGroup(
255     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
256     .addGroup(layout.createSequentialGroup()
257         .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
258             .add(layout.createSequentialGroup()
259                 .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
260                     .add(layout.createSequentialGroup()
261                         .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
262                             .add(layout.createSequentialGroup()
263                                 .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
264                                     .add(layout.createSequentialGroup()

```

```

265         GroupLayout.Alignment.LEADING)
266         .addComponent(jLabel12)
267         .addComponent(jLabel13)
268         .addComponent(jLabel15)
269         .addComponent(jLabel10)
270         .addComponent(jLabel19)
271         .addComponent(jLabel14)
272         .addComponent(jLabel13))
273     .addGap(29, 29, 29)
274     .addGroup(layout.createParallelGroup(javax.swing.
275         GroupLayout.Alignment.LEADING)
276         .addComponent(jTextField5, javax.swing.GroupLayout.
277             DEFAULT_SIZE, 69, Short.MAX_VALUE)
278         .addGroup(layout.createParallelGroup(javax.swing.
279             GroupLayout.Alignment.LEADING, false)
280             .addComponent(jTextField4)
281             .addComponent(jTextField3)
282             .addComponent(jTextField2)
283             .addComponent(jTextField7, javax.swing.
284                 GroupLayout.PREFERRED_SIZE, 69, javax.swing.
285                 GroupLayout.PREFERRED_SIZE)
286             .addComponent(jTextField8, javax.swing.
287                 GroupLayout.PREFERRED_SIZE, 69, javax.swing.
288                 GroupLayout.PREFERRED_SIZE))
289         .addComponent(jTextField9, javax.swing.GroupLayout.
290             DEFAULT_SIZE, 69, Short.MAX_VALUE)))
291     .addGap(28, 28, 28)
292     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
293         Alignment.LEADING)
294         .addComponent(jScrollPane3, javax.swing.GroupLayout.
295             DEFAULT_SIZE, 237, Short.MAX_VALUE)
296         .addComponent(jLabel16)
297         .addGroup(layout.createSequentialGroup())
298             .addGroup(layout.createParallelGroup(javax.swing.
299                 GroupLayout.Alignment.LEADING)
300                 .addComponent(jLabel18)
301                 .addComponent(jLabel11)
302                 .addComponent(jLabel12))
303             .addGap(14, 14, 14)
304             .addGroup(layout.createParallelGroup(javax.swing.
305                 GroupLayout.Alignment.LEADING)
306                 .addComponent(jTextField6, javax.swing.GroupLayout.
307                     Alignment.TRAILING, javax.swing.GroupLayout.
308                     DEFAULT_SIZE, 136, Short.MAX_VALUE)
309                 .addComponent(jProgressBar1, javax.swing.GroupLayout.
310                     DEFAULT_SIZE, 136, Short.MAX_VALUE)
311                 .addComponent(jProgressBar2, javax.swing.GroupLayout.
312                     Alignment.TRAILING, javax.swing.GroupLayout.
313                     DEFAULT_SIZE, 136, Short.MAX_VALUE))))
314     .addComponent(jLabel17, javax.swing.GroupLayout.DEFAULT_SIZE, 476,
315         Short.MAX_VALUE))
316     .addContainerGap()
317 );
318 layout.setVerticalGroup(
319     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
320     .addGroup(layout.createSequentialGroup()
321         .addContainerGap()
322         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
323             LEADING)
324             .addGroup(layout.createSequentialGroup()
325                 .addComponent(jButton1)
326                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
327                     UNRELATED)
328                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
329                     Alignment.BASELINE)
330                     .addComponent(jTextField7, javax.swing.GroupLayout.
331                         PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
332                         javax.swing.GroupLayout.PREFERRED_SIZE)
333                     .addComponent(jLabel19))
334                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.

```

```

311         RELATED)
312     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
313         Alignment.BASELINE)
314         .addComponent(jTextField8, javax.swing.GroupLayout.
315             PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
316             javax.swing.GroupLayout.PREFERRED_SIZE)
317         .addComponent(jLabel10))
318     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
319         RELATED)
320     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
321         Alignment.BASELINE)
322         .addComponent(jLabel12)
323         .addComponent(jTextField2, javax.swing.GroupLayout.
324             PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
325             javax.swing.GroupLayout.PREFERRED_SIZE))
326     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
327         RELATED)
328     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
329         Alignment.BASELINE)
330         .addComponent(jLabel13)
331         .addComponent(jTextField3, javax.swing.GroupLayout.
332             PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
333             javax.swing.GroupLayout.PREFERRED_SIZE))
334     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
335         RELATED)
336     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
337         Alignment.BASELINE)
338         .addComponent(jLabel15)
339         .addComponent(jTextField4, javax.swing.GroupLayout.
340             PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
341             javax.swing.GroupLayout.PREFERRED_SIZE)
342         .addComponent(jLabel18)
343         .addComponent(jTextField6, javax.swing.GroupLayout.
344             PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
345             javax.swing.GroupLayout.PREFERRED_SIZE)))
346     .addGroup(layout.createSequentialGroup())
347     .addComponent(jLabel6)
348     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
349         RELATED)
350     .addComponent(jScrollPane3, javax.swing.GroupLayout.
351         PREFERRED_SIZE, 106, javax.swing.GroupLayout.PREFERRED_SIZE)
352     ))
353     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
354     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
355         BASELINE)
356         .addComponent(jLabel14)
357         .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
358             javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.
359             PREFERRED_SIZE)
360         .addComponent(jLabel11)
361         .addComponent(jProgressBar1, javax.swing.GroupLayout.PREFERRED_SIZE,
362             javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.
363             PREFERRED_SIZE))
364     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
365     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
366         BASELINE)
367         .addComponent(jLabel13)
368         .addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED_SIZE,
369             javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.
370             PREFERRED_SIZE)
371         .addComponent(jLabel12)
372         .addComponent(jProgressBar2, javax.swing.GroupLayout.PREFERRED_SIZE,
373             javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.
374             PREFERRED_SIZE))
375     .addGap(18, 18, 18)
376     .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 329,
377         Short.MAX_VALUE)
378     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
379     .addComponent(jLabel7)
380     .addContainerGap()

```



```

349     );
350
351     pack();
352 }// </editor-fold>//GEN-END: initComponents
353
354 /**
355  * Acció d'activar el sensor d'humitat en polsar el botó de la GUI
356  * @param evt    Event captat pel botó
357  */
358 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:
359     event_jButton1ActionPerformed
360     S.activarSensor();
361 } //GEN-LAST: event_jButton1ActionPerformed
362
363 private void jTextField7ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST
364     :event_jTextField7ActionPerformed
365     // TODO add your handling code here:
366 } //GEN-LAST: event_jTextField7ActionPerformed
367
368 private void jTextField8ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST
369     :event_jTextField8ActionPerformed
370     // TODO add your handling code here:
371 } //GEN-LAST: event_jTextField8ActionPerformed
372
373 // Variables declaration - do not modify //GEN-BEGIN: variables
374 private javax.swing.JButton jButton1;
375 private javax.swing.JLabel jLabel10;
376 private javax.swing.JLabel jLabel11;
377 private javax.swing.JLabel jLabel12;
378 private javax.swing.JLabel jLabel13;
379 private javax.swing.JLabel jLabel2;
380 private javax.swing.JLabel jLabel3;
381 private javax.swing.JLabel jLabel4;
382 private javax.swing.JLabel jLabel5;
383 private javax.swing.JLabel jLabel6;
384 private javax.swing.JLabel jLabel7;
385 private javax.swing.JLabel jLabel8;
386 private javax.swing.JLabel jLabel9;
387 private javax.swing.JProgressBar jProgressBar1;
388 private javax.swing.JProgressBar jProgressBar2;
389 private javax.swing.JScrollPane jScrollPane2;
390 private javax.swing.JScrollPane jScrollPane3;
391 private javax.swing.JTable jTable1;
392 private javax.swing.JTextField jTextField2;
393 private javax.swing.JTextField jTextField3;
394 private javax.swing.JTextField jTextField4;
395 private javax.swing.JTextField jTextField5;
396 private javax.swing.JTextField jTextField6;
397 private javax.swing.JTextField jTextField7;
398 private javax.swing.JTextField jTextField8;
399 private javax.swing.JTextField jTextField9;
400 private javax.swing.JTextPane jTextPane1;
401 // End of variables declaration //GEN-END: variables
402 }

```

#### Codi A.4: (Simulador) Model.java

```

1 package ppstop_simulador;
2
3 import java.util.Observable;
4 import java.util.Vector;
5
6 /**
7  * Classe que conté tots els paràmetres del simulador. Actua com a Model dins del patró
8  * MVC (Model-View-Controller)
9  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
10  * @version 1.0 (Juliol de 2009)
11  */

```

```

11 public class Model extends Observable {
12
13     /** Flag d'inici del dispositiu PPStop */
14     private int flag_inicial = 1;
15     /** Nombre de terminals PPStop associats al dispositiu PPStop */
16     private int num_dispositius_ppstop = 0;
17     /** Nombre d'intents de connexió del dispositiu PPStop */
18     private int intents_connexio = 0;
19     /** Nombre de connexions establertes pel dispositiu PPStop */
20     private int connexions_establertes = 0;
21     /** Nombre de peticions de canvi de sensibilitat rebudes pel dispositiu PPStop */
22     private int peticions_sensibilitat = 0;
23     /** Nombre de peticions de buscar terminals rebudes pel dispositiu PPStop */
24     private int peticions_busqueda = 0;
25     /** Nombre de peticions de canvi d'ordre dels terminals associats */
26     private int peticions_canviordre = 0;
27     /** Nivell de sensibilitat del sensor d'humitat del dispositiu PPStop (1-5) */
28     private int sensibilitat_sensor = 3;
29     /** Estat de la bateria del dispositiu PPStop (%) */
30     private int estat_bateria = 100;
31     /** Adreça bluetooth del dispositiu PPStop */
32     private String MACSimulador = null;
33     /** Status del dispositiu PPStop */
34     private String status = null;
35     /** Indica si el botó d'activació del sensor d'humitat està activat */
36     private boolean SensorButton = false;
37     /* Vector de dispositius bluetooth (terminals mòbils PPStop) trobats */
38     private Vector<Dispositiu> Dispositius = new Vector();
39
40     /**
41      * Canvia l'estat del botó d'activació del sensor d'humitat
42      * @param b      Estat del botó (booleà)
43      */
44     public void enableSensorButton(boolean b){
45         SensorButton = b;
46         setChanged();
47         notifyObservers();
48     }
49
50     /**
51      * Obté l'estat del botó d'activació del sensor d'humitat
52      * @return Estat del botó d'activació (booleà)
53      */
54     public boolean getSensorButton(){
55         return SensorButton;
56     }
57
58     /**
59      * Canvia l'status del dispositiu PPStop
60      * @param text  Text que defineix l'status
61      */
62     public void setStatus(String text){
63         status = text;
64         setChanged();
65         notifyObservers();
66     }
67
68     /**
69      * Obté l'status del dispositiu PPStop
70      * @return Text que defineix l'status
71      */
72     public String getStatus(){
73         return status;
74     }
75
76     /**
77      * Defineix l'adreça bluetooth del dispositiu PPStop
78      * @param Addr  Adreça bluetooth
79      */
80     public void setMACSimulador(String Addr){

```

```

81         MACSimulador = Addr;
82         setChanged();
83         notifyObservers();
84     }
85
86     /**
87     * Obté l'adreça bluetooth del dispositiu PPStop
88     * @return Adreça bluetooth
89     */
90     public String getMACSimulador(){
91         return MACSimulador;
92     }
93
94     /**
95     * Modifica el flag inicial del dispositiu PPStop
96     * @param flag      Valor del flag
97     */
98     public void setFlagInicial(Integer flag){
99         flag_inicial = flag;
100         setChanged();
101         notifyObservers();
102     }
103
104     /**
105     * Obté el flag inicial del dispositiu PPStop
106     * @return Valor del flag
107     */
108     public Integer getFlagInicial(){
109         return flag_inicial;
110     }
111
112     /**
113     * Incrementa el nombre d'intents de connexió
114     */
115     public void incrementIntentsConnexio(){
116         intents_connexio ++;
117         setChanged();
118         notifyObservers();
119     }
120
121     /**
122     * Obté els nombre d'intents de connexió
123     * @return Nombre d'intents de connexió
124     */
125     public Integer getIntentsConnexio(){
126         /* Obtenir els Intents de Connexió */
127         return intents_connexio;
128     }
129
130     /**
131     * Incrementa el nombre de connexions establertes
132     */
133     public void incrementConnexionsEstablertes(){
134         connexions_establertes ++;
135         setChanged();
136         notifyObservers();
137     }
138
139     /**
140     * Obté el nombre de connexions establertes
141     * @return Nombre de connexions establertes
142     */
143     public Integer getConnexionsEstablertes(){
144         return connexions_establertes;
145     }
146
147     /**
148     * Modifica el nivell de sensibilitat del sensor d'humitat
149     * @param num      Nivell de sensibilitat (1-5)
150     */

```

```

151 public void setSensibilitatSensor(int num){
152     sensibilitat_sensor = num;
153     setChanged();
154     notifyObservers();
155 }
156
157 /**
158  * Obté el nivell de sensibilitat del sensor d'humitat
159  * @return Nivell de sensibilitat (1-5)
160  */
161 public Integer getSensibilitatSensor(){
162     return sensibilitat_sensor;
163 }
164
165 /**
166  * Obté el nivell de sensibilitat del sensor d'humitat en format TXT per a TX
167  * @return Nivell de sensibilitat (1-5)
168  */
169 public String getSensibilitatSensorTXT(){
170     return String.valueOf(sensibilitat_sensor);
171 }
172
173 /**
174  * Incrementa el nombre de peticions de canvi de sensibilitat
175  */
176 public void incrementPeticionsSensibilitat(){
177     peticions_sensibilitat ++;
178     setChanged();
179     notifyObservers();
180 }
181
182 /**
183  * Obté el nombre de peticions de canvi de sensibilitat
184  * @return Nombre de peticions
185  */
186 public Integer getPeticionsSensibilitat(){
187     return peticions_sensibilitat;
188 }
189
190 /**
191  * Incrementa el nombre de peticions de búsqueda de terminals
192  */
193 public void incrementPeticionsBusqueda(){
194     peticions_busqueda ++;
195     setChanged();
196     notifyObservers();
197 }
198
199 /**
200  * Obté el nombre de peticions de búsqueda de terminals PPStop
201  * @return Nombre de peticions
202  */
203 public Integer getPeticionsBusqueda(){
204     return peticions_busqueda;
205 }
206
207 /**
208  * Incrementa el nombre de peticions de canvi d'ordre dels terminals PPStop
    associats
209  */
210 public void incrementPeticionsCanviOrdre(){
211     peticions_canviordre ++;
212     setChanged();
213     notifyObservers();
214 }
215
216 /**
217  * Obté el nombre de peticions de canvi d'ordre dels terminals PPStop associats
218  * @return Nombre de peticions
219  */

```

```

220 public Integer getPeticionsCanviOrdre(){
221     return petitions_canviordre;
222 }
223
224 /**
225  * Decrementa l'estat de la bateria
226  * @param num      Decrement de bateria (%)
227  */
228 public void decrementEstatBateria(int num){
229     estat_bateria -= num;
230     if (estat_bateria<0) estat_bateria=0;
231     setChanged();
232     notifyObservers();
233 }
234
235 /**
236  * Obté l'estat de la bateria
237  * @return Estat de la bateria (%)
238  */
239 public Integer getEstatBateria(){
240     return estat_bateria;
241 }
242
243 /**
244  * Obté l'estat de la bateria en format TXT de 3 dígits per a TX
245  * @return Estat de la bateria (3 dígits)
246  */
247 public String getEstatBateriaTXT(){
248     String estat_bateria_txt = String.valueOf(estat_bateria);
249     if (estat_bateria<100) estat_bateria_txt = "0" + estat_bateria_txt;
250     if (estat_bateria<10) estat_bateria_txt = "0" + estat_bateria_txt;
251     return estat_bateria_txt;
252 }
253
254 /**
255  * Incrementa el nombre de terminals PPStop trobats
256  */
257 public void incrementNumDispositiusPPStopTrobats(){
258     num_dispositius_ppstop ++;
259     setChanged();
260     notifyObservers();
261 }
262
263 /**
264  * Estableix el nombre de terminals PPStop trobats
265  * @param num      Nombre de terminals
266  */
267 public void setNumDispositiusPPStopTrobats(int num){
268     num_dispositius_ppstop = num;
269 }
270
271 /**
272  * Obté el nombre de terminals PPStop trobats
273  * @return Nombre de terminals
274  */
275 public int getNumDispositiusPPStopTrobats(){
276     return num_dispositius_ppstop;
277 }
278
279 /**
280  * Estableix el vector de terminals PPStop trobats
281  * @param V        Vector de terminals PPStop
282  */
283 public void setDispositiusPPStopTrobats(final Vector<Dispositiu> V){
284     Dispositius = V;
285     setChanged();
286     notifyObservers();
287 }
288
289 /**

```

```

290     * Obté el vector de terminals PPStop trobats
291     * @return Vector de terminals PPStop
292     */
293     public Vector<Dispositiu> getDispositiusPPStopTrobats(){
294         return Dispositius;
295     }
296 }
297 }

```

### Codi A.5: (Simulador) Simulador.java

```

1 package ppstop_simulador;
2
3 /**
4  * Classe principal del simulador
5  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
6  * @version 1.0 (Juliol de 2009)
7  */
8 public class Simulador{
9
10     /** Objecte per a la Comunicació Sèrie amb el transmissor BT */
11     private static ComunicacioSerie CS;
12     /** Unitat de Control: lògica del dispositiu PPStop simulat */
13     private static UnitatControl UC;
14     /** Model de dades (patró MVC) */
15     private static Model M;
16     /** GUI del simulador (patró MVC) */
17     private static GUI V;
18
19     /**
20      * Crea una instància de Simulador
21      */
22     public Simulador(){}
23
24     /**
25      * Obte el port sèrie i inicia el protocol
26      */
27     public void iniciar(){
28         CS.obrirPort("COM6");
29         UC.actuar();
30     }
31
32     /**
33      * Tanca el port sèrie i surt del simulador
34      */
35     public void finalitzar(){
36         CS.tancarPort();
37         System.exit(0);
38     }
39
40     /**
41      * Envia una comanda AT al transmissor BT
42      * @param comandaAT Comanda AT a enviar
43      */
44     public void enviarAT(String comandaAT){
45         CS.enviarAT(comandaAT);
46     }
47
48     /**
49      * Comunica a la Unitat de Control la resposta obtinguda pel transmissor BT
50      * @param resposta Resposta obtinguda
51      */
52     public void setResposta(String resposta){
53         UC.setResposta(resposta);
54         UC.actuar();
55     }
56
57     /**
58      * Actua segons l'estat del protocol

```

```

59     */
60     public void activarSensor(){
61         UC.activarSensor();
62     }
63
64     /**
65     * Afegeix una línia de text al LOG de pantalla
66     * @param s      Línia de text a afegir al LOG
67     */
68     public void log(String s){
69         V.afegirLog("--> " + s);
70     }
71
72     /**
73     * Programa principal del simulador
74     * @param args      Arguments del simulador
75     */
76     public static void main(String[] args) {
77
78         final Simulador S = new Simulador();
79         M = new Model();
80         CS = new ComunicacioSerie(S);
81         UC = new UnitatControl(S,M);
82         try {
83             javax.swing.SwingUtilities.invokeLater(new Runnable() {
84                 public void run() {
85                     V = new GUI(S,M);
86                     V.setVisible(true);
87                 }
88             });
89         } catch (Exception e) {
90             System.out.println("Error: " + e);
91         }
92         M.addObserver(V);
93         S.iniciar();
94     }
95 }
96 }

```

#### Codi A.6: (Simulador) UnitatControl.java

```

1 package ppstop_simulador;
2
3 import java.util.Vector;
4
5
6 /**
7  * Classe que conté la Unitat de Control del dispositiu PPStop simulat
8  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
9  * @version 1.0 (Juliol de 2009)
10  */
11 public class UnitatControl {
12
13     /** Referència al Simulador */
14     private Simulador S;
15     /** Referència al Model de dades */
16     private Model M;
17
18     /** Estat actual de la màquina d'estats */
19     private int estat;
20     /** Estat futur de la màquina d'estats */
21     private int estatFutur;
22     /** Nivell de sensibilitat rebut */
23     private int sensibilitatRebuda;
24     /** Resposta rebuda del transmissor bluetooth */
25     private String resposta;
26     /** Vector de terminals PPStop */
27     private Vector<Dispositiu> Dispositius = new Vector();
28     /** Indica si el terminal que s'està analitzant és un terminal PPStop */

```

```

29 private boolean dispositiu_PPStop;
30 /** Indica si el terminal que s'està analitzant és un terminal PPStop autoritzat */
31 private boolean dispositiu_PPStop_autoritzat;
32 /** Variable per al control del temps total de connexió */
33 private long tinicial_connexio;
34 /** Variable per al control del temps total de connexió */
35 private long tfinal_connexio;
36 /** Variable auxiliar per als bucles */
37 private int i;
38
39 /** Estat d'obtenció de l'adreça bluetooth del dispositiu (transmissor bluetooth) */
40 private static final int ESTAT_GETADDRESS = 0;
41 /** Estat de reset del dispositiu PPStop (transmissor bluetooth) */
42 private static final int ESTAT_RESET = 1;
43 /** Estat de definició de la seguretat de les comunicacions bluetooth */
44 private static final int ESTAT_DEFINIRSEGURETAT = 2;
45 /** Estat d'inicialitzar els pins del transmissor */
46 private static final int ESTAT_INIPINS = 3;
47 /** Estat de definir el nom del dispositiu bluetooth */
48 private static final int ESTAT_POSARNOM = 4;
49 /** Estat de búsqueda de terminals en àrea de cobertura */
50 private static final int ESTAT_BUSQUEDATERMINALS = 5;
51 /** Estat d'espera de resposta a la búsqueda de terminals en àrea de cobertura */
52 private static final int ESTAT_BUSQUEDATERMINALS_RESP = 6;
53 /** Estat de búsqueda de servei PPStop en els terminals trobats */
54 private static final int ESTAT_BUSQUEDASERVEIS = 7;
55 /** Estat d'espera de resposta a la búsqueda de servei PPStop en els terminals
    trobats */
56 private static final int ESTAT_BUSQUEDASERVEIS_RESP = 8;
57 /** Estat de comprovació de servei PPStop autoritzat */
58 private static final int ESTAT_COMPROVARSERVEI = 9;
59 /** Estat d'espera de resposta (1) a la comprovació de servei PPStop autoritzat */
60 private static final int ESTAT_COMPROVARSERVEI_RESP1 = 10;
61 /** Estat d'espera de resposta (2) a la comprovació de servei PPStop autoritzat */
62 private static final int ESTAT_COMPROVARSERVEI_RESP2 = 11;
63 /** Estat d'espera d'event (activar el sensor d'humitat) */
64 private static final int ESTAT_ESPERAREVENT = 15;
65 /** Estat de canviar la sensibilitat del sensor d'humitat del dispositiu PPStop */
66 private static final int ESTAT_CANVIARSENSIBILITAT = 22;
67 /** Estat d'establir una connexió d'alarma als terminals PPStop associats */
68 private static final int ESTAT_ESTABLIRCONNEXIO = 30;
69 /** Estat d'espera de resposta (1) a la connexió d'alarma als terminals PPStop
    associats */
70 private static final int ESTAT_ESTABLIRCONNEXIO_RESP1 = 31;
71 /** Estat d'espera de resposta (2) a la connexió d'alarma als terminals PPStop
    associats */
72 private static final int ESTAT_ESTABLIRCONNEXIO_RESP2 = 32;
73
74 /** Comanda AT d'obtenció de l'adreça bluetooth */
75 private static final String COMANDA_GETADDRESS = "AT+JRBD";
76 /** Comanda AT de reset */
77 private static final String COMANDA_RESET = "AT+JRES";
78 /** Comanda AT d'inicialització dels pins */
79 private static final String COMANDA_INIPINS = "AT+JGPC=0000,0000,0000,0000,0000";
80 /** Comanda AT d'establiment de seguretat: sense autenticació ni encriptació */
81 private static final String COMANDA_SEGURETAT = "AT+JSEC=1,1,1,04,1111";
82 /** Comanda AT d'establiment del nom del dispositiu */
83 private static final String COMANDA_POSARNOM = "AT+JSLN=10,PPSTOP_XIP";
84 /** Comanda AT d'inici de búsqueda de terminals */
85 private static final String COMANDA_BUSQUEDATERMINALS = "AT+JDDS=0";
86 /** Comanda AT (parcial 1) d'inici de búsqueda de serveis en un determinat terminal
    */
87 private static final String COMANDA_BUSQUEDASERVEIS_1 = "AT+JSDS=";
88 /** Comanda AT (parcial 2) d'inici de búsqueda de serveis en un determinat terminal
    */
89 private static final String COMANDA_BUSQUEDASERVEIS_2 = ",1101";
90 /** Comanda AT (parcial) per a enviar l'ordre de comprovar si un terminal PPStop és
    un terminal autoritzat */
91 private static final String COMANDA_DATA_COMPROVAR = "AT+JSDA=005,C";
92 /** Comanda AT (parcial) per a enviar l'ordre d'alarmar */

```



```

93 private static final String COMANDA_DATA_ALERTAR = "AT+JSDA=005,A";
94 /** Comanda AT (parcial) per a iniciar una connexió bluetooth */
95 private static final String COMANDA_CONNEXIO = "AT+JCCR=";
96
97 /**
98  * Crea una instància de UnitatControl
99  * @param S      Referència al Simulador
100  * @param M      Referència al Model de dades
101  */
102 public UnitatControl(Simulador S, Model M){
103     this.S = S;
104     this.M = M;
105     estat = ESTAT_GETADRESS;
106 }
107
108 /**
109  * Estableix la resposta obtinguda pel port sèrie
110  * @param resposta      Resposta obtinguda
111  */
112 public void setResposta(String resposta){
113     this.resposta = resposta;
114 }
115
116 /**
117  * Activa el sensor d'humitat actuant conseqüentment segons l'estat del protocol
118  */
119 public void activarSensor(){
120     estat = (M.getFlagInicial()==1) ? ESTAT_BUSQUEDATERMINALS :
121         ESTAT_ESTABLIRCONNEXIO;
122     resposta = null;
123     M.enableSensorButton(false);
124     i=0;
125     actuar();
126 }
127
128 /**
129  * Màquina d'estats: canvia d'estat segons la resposta obtinguda i actua en
130  * conseqüència
131  */
132 public void actuar() {
133     switch(estat){
134         case ESTAT_GETADRESS:
135             if(resposta==null){
136                 M.setStatus("Iniciant la configuració del xip...");
137                 M.enableSensorButton(false);
138                 S.enviarAT(COMANDA_GETADRESS);
139             } else {
140                 String localaddress=(resposta.split("=")[1]).split(",")[0];
141                 M.setMACSimulador(localaddress);
142                 resposta=null;
143                 estat++;
144                 actuar();
145             }
146             break;
147         case ESTAT_RESET:
148             if(resposta==null){
149                 S.enviarAT(COMANDA_RESET);
150             } else if (resposta.equals("ROK")){
151                 resposta=null;
152                 estat++;
153                 actuar();
154             }
155             break;
156         case ESTAT_INIPINS:
157             if(resposta==null){
158                 S.enviarAT(COMANDA_INIPINS);
159             }
160     }

```

```

161         } else if (resposta.equals("OK")){
162             resposta=null;
163             estat++;
164             actuar();
165         }
166         break;
167
168     case ESTAT_DEFINIRSEGURETAT:
169         if(resposta==null){
170             S.enviarAT(COMANDA_SEGURETAT);
171         } else if (resposta.equals("OK")){
172             resposta=null;
173             estat++;
174             actuar();
175         }
176         break;
177
178     case ESTAT_POSARNOM:
179         if(resposta==null){
180             S.enviarAT(COMANDA_POSARNOM);
181         } else if (resposta.equals("OK")){
182             resposta=null;
183             estat++;
184             actuar();
185         }
186         break;
187
188     case ESTAT_BUSQUEDATERMINALS:
189         if(resposta==null){
190             M.setStatus("Iniciant la búsqueda de dispositius...");
191             Dispositius.clear();
192             M.setDispositiusPPStopTrobats(Dispositius);
193             M.setNumDispositiusPPStopTrobats(0);
194             i=0;
195             S.enviarAT(COMANDA_BUSQUEDATERMINALS);
196             M.decrementEstatBateria(2);
197         } else if (resposta.equals("OK")){
198             resposta=null;
199             estat++;
200         }
201         break;
202
203     case ESTAT_BUSQUEDATERMINALS_RESP:
204         if(resposta.split("=")[0].equals("+RDDSCNF")){
205             M.setStatus("Iniciant la búsqueda de serveis PPStop...");
206             resposta=null;
207             estat++;
208             actuar();
209         } else {
210             String dispAddr=(resposta.split("=")[1]).split(",")[0];
211             String dispNom=(resposta.split("=")[1]).split(",")[1];
212             Dispositius.add(new Dispositiu(-1,dispNom,dispAddr,null));
213         }
214         break;
215
216     case ESTAT_BUSQUEDASERVEIS:
217         if(resposta==null){
218             dispositiu_PPStop = false;
219             S.enviarAT(COMANDA_BUSQUEDASERVEIS_1 + Dispositius.get(i).
220                 getDispAddr() + COMANDA_BUSQUEDASERVEIS_2);
221             M.decrementEstatBateria(1);
222         } else if (resposta.equals("OK")){
223             resposta=null;
224             estat++;
225         }
226         break;
227
228     case ESTAT_BUSQUEDASERVEIS_RESP:
229         if (resposta.split("=")[0].equals("+RSDSCNF")){
230             resposta=null;

```

```

230         if (dispositiu_PPStop){
231             i++;
232         } else {
233             Dispositius.remove(i);
234         }
235         if (Dispositius.size() == i){
236             i=0;
237             M.setDispositiusPPStopTrobats(Dispositius);
238             estat++;
239         } else {
240             estat--;
241         }
242         actuar();
243     } else if (!resposta.equals("+RRSW")) {
244         String servNom=(resposta.split("=")[1]).split(",")[0];
245         if (servNom.equalsIgnoreCase("PPStop")){
246             String servChann=(resposta.split("=")[1]).split(",")[1];
247             Dispositius.get(i).setDispChan(servChann);
248             dispositiu_PPStop = true;
249         }
250     }
251     break;
252
253     case ESTAT_COMPROVARSERVEI:
254         if(resposta==null){
255             if (Dispositius.size()==i){
256                 i=0;
257                 M.setStatus("Esperant la recepció d'events...");
258                 M.enableSensorButton(true);
259                 estat = ESTAT_ESPERAREVENT;
260             } else {
261                 M.setStatus("Verificant la configuració dels serveis...");
262                 dispositiu_PPStop_autoritzat = false;
263                 S.enviarAT(COMANDA_CONNEXIO + Dispositius.get(i).getDispAddr() +
264                     ", " + Dispositius.get(i).getDispChan());
265                 M.decrementEstatBateria(1);
266                 M.incrementIntentsConnexio();
267             }
268         } else if (resposta.equals("OK")){
269             resposta=null;
270             estat++;
271         }
272         break;
273
274     case ESTAT_COMPROVARSERVEI_RESP1:
275         if (resposta.split("=")[0].equals("+RCCRCNF")){
276             String statusConnexio = (resposta.split("=")[1]).split(",")[1];
277             if (statusConnexio.equals("1")){
278                 S.log("Connexió NO establerta...");
279                 estat--;
280             } else {
281                 S.log("Connexió establerta...");
282                 estat++;
283             }
284             resposta=null;
285             actuar();
286         } else if (resposta.equals("+RDII")){
287             M.incrementConnexionsEstablertes();
288             resposta=null;
289             estat--;
290             actuar();
291         } else if (resposta.equals("OK")){
292             resposta=null;
293             estat++;
294         }
295         break;
296
297     case ESTAT_COMPROVARSERVEI_RESP2:
298         if(resposta==null){
299             S.enviarAT(COMANDA_DATA_COMPROVAR + M.getEstatBateriaTXT() + M.

```

```

309         getSensibilitatSensorTXT());
300     } else if (resposta.split("=")[0].equals("+RDAI")){
301         String respTerm = (resposta.split("=")[1]).split(",")[1];
302         if (respTerm.equals("OK")){
303             S.log("El servei s'ha verificat i ha sigut ACCEPTAT...");
304             dispositiu_PPStop_autoritzat = true;
305             M.incrementNumDispositiusPPStopTrobats();
306             Dispositius.get(i).setDispOrd(M.getNumDispositiusPPStopTrobats()
307             );
308             M.setFlagInicial(0);
309             i++;
310         }
311     } else if (resposta.equals("+RDII")){
312         if (dispositiu_PPStop_autoritzat == false){
313             S.log("El servei s'ha verificat i ha sigut REBUTJAT...");
314             Dispositius.remove(i);
315         }
316         M.incrementConnexionsEstablertes();
317         resposta=null;
318         estat = ESTAT_COMPROVARSERVEI;
319         actuar();
320     }
321     break;
322
323 case ESTAT_ESPERAREVENT:
324     break;
325
326 case ESTAT_CANVIARSENSIBILITAT:
327     // Accions per canviar la sensibilitat del micro
328     // ...
329     M.setSensibilitatSensor(sensibilitatRebuda);
330     estat = ESTAT_ESPERAREVENT;
331     M.setStatus("Esperant la recepció d'events...");
332     break;
333
334 case ESTAT_ESTABLIRCONNEXIO:
335     if(resposta==null){
336         if (Dispositius.size()==i){
337             M.enableSensorButton(true);
338             i=0;
339             M.setStatus("Esperant la recepció d'events...");
340             estat = ESTAT_ESPERAREVENT;
341         } else {
342             M.setStatus("Establint connexió amb els dispositius...");
343             S.enviarAT(COMANDA_CONNEXIO + Dispositius.get(i).getDispAddr() +
344             ", " + Dispositius.get(i).getDispChan());
345             M.decrementEstatBateria(1);
346             if (i==0){
347                 tinicial_connexio = System.currentTimeMillis();
348             }
349             i++;
350             M.incrementIntentsConnexio();
351         }
352     } else if (resposta.equals("OK")){
353         resposta=null;
354         estat++;
355     }
356     break;
357
358 case ESTAT_ESTABLIRCONNEXIO_RESPI:
359     if (resposta.split("=")[0].equals("+RCCRCNF")){
360         String statusConnexio = (resposta.split("=")[1]).split(",")[1];
361         if (statusConnexio.equals("1")){
362             S.log("Connexió NO establerta...");
363             resposta=null;
364             estat--;
365             actuar();
366         } else {
367             S.log("Connexió establerta...");
368             S.enviarAT(COMANDA_DATA_ALERTAR + M.getEstatBateriaTXT() + M.

```

```

366         getSensibilitatSensorTXT());
367     }
368     } else if (resposta.equals("OK")){
369         resposta=null;
370         estat++;
371     }
372     break;
373
374     case ESTAT_ESTABLIRCONNEIXIO_RESP2:
375         if (resposta.split("=")[0].equals("+RDAI")){
376             String msgRebut = (resposta.split("=")[1]).split(",")[1];
377             if (msgRebut.equals("OK")){
378                 S.log("Alarma servida correctament!");
379                 estatFutur = ESTAT_ESTABLIRCONNEIXIO;
380             } else if (msgRebut.equals("B")){
381                 S.log("Nou event rebut: búsqueda de dispositius");
382                 M.incrementPetitionsBusqueda();
383                 estatFutur = ESTAT_BUSQUEDATERMINALS;
384                 i=0;
385             } else if (msgRebut.substring(0,1).equals("S")){
386                 sensibilitatRebuda = Integer.valueOf(msgRebut.substring(1,2));
387                 S.log("Nou event rebut: canvi de sensibilitat del sensor (nivell
388                     " + sensibilitatRebuda + ")");
389                 M.setStatus("Canviant la sensibilitat del sensor...");
390                 M.incrementPetitionsSensibilitat();
391                 estatFutur = ESTAT_CANVIARSENSIBILITAT;
392             }
393         } else if (resposta.equals("+RDII")){
394             resposta=null;
395             tfinal_connexio = System.currentTimeMillis();
396             S.log("Temps de connexió: " + (tfinal_connexio - tinicial_connexio)
397                 + "ms.");
398             M.incrementConnexionsEstablertes();
399             estat = estatFutur;
400             M.enableSensorButton(true);
401             actuar();
402         }
403     }
404 }
405 }

```

### Codi A.7: (Simulador) Vista.java

```

1 package ppstop_simulador;
2
3 import java.util.Observer;
4
5 /**
6  * Interficie d'un observador per a la Vista (GUI) del patró MVC
7  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
8  * @version 1.0 (Juliol de 2009)
9  */
10 public interface Vista extends Observer { }

```



## Apèndix B

### Dispositiu electrònic





## B.1 Transmisor Bluetooth Infineon eUniStone PBA31308/2 SPP-AT



# eUniStone PBA31308/2 SPP-AT

## BlueMoon® Universal Family

### Extended to Embedded Data Transfer Applications

THE COMPLETE SPP SYSTEM MODULE, eUniStone is a fully integrated Bluetooth® device including stack and Serial Port Profile for data transfer applications. With an easy command interface on top enabling an easy in-design and minimal Host requirements.

The ready to use Bluetooth System Module offers a qualified solution for accelerated time-to-market. The high level of integration results into a competitive module footprint providing an optimized pad layout for easy design-in. The eUniStone has been designed with the focus on low cost, long range and easy design-in.

The high performing eUniStone offers outstanding range and efficient utilization of low power modes. It is compliant to BT 2.0 plus enhanced data rate and is equipped with a powerful ARM7 processor.

The low-cost system requires no external components except the antenna. The small module offers a small solution footprint. Reference design and design kit are available for quick and easy design-in.

The optimized system partitioning with patching capabilities of ROM code together with applications and complimentary profiles loaded from external memory provides a highly flexible and upgradeable solution.

The complete eUniStone SYSTEM SW includes the Serial Port Profile (SPP) with an extended AT command-set, supporting both Device A and Device B role configuration. HID profile and other applications can be offered as Customized Software on request.

#### Applications

- Consumer, Automotive & Industrial Applications
- Data transfer & synchronization
- Remote control
- Sensors

#### Main Features

- Complete Bluetooth system device
- Pre qualified Bluetooth product v2.0 + EDR
  - Enhanced data rate
  - Adaptive frequency hopping
  - Fast connection setup
  - On-module Bluetooth reference crystal
  - On-module E²PROM for Bluetooth device data and SW

#### Main Features

- SPP profile implementation
- SPP Device A and B support
- SW application upgrade over UART
- GPIO Support for up to 8 GPIO
- Bluetooth Power Class 2
- High RF sensitivity (-86dBm @ 0.01% BER)
- RoHS compliant

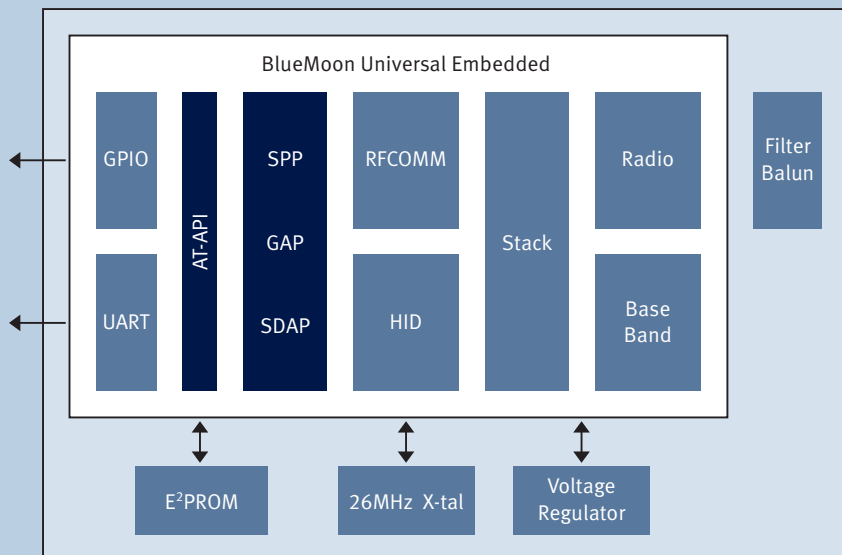
#### Key Benefits

- Complete system solution including BT stack with SPP, GAP, SDAP profile and AT command set
- One-Stop-Shop at Infineon
  - Module directly from IC and software manufacturer
  - Optimized value-chain (lead time, cost, responsibility)
- No external components needed except antenna
- Pre-qualified for Bluetooth, FCC and R&TTE
- Patching capability offering large flexibility on
  - Enhancement of existing functionalities
- Bug fixing
- High performance processor system enabling minimal requirements of external host solution
- Specific Production mode
- Relaxed pad pitch - 1.2mm - easy design
- Infineon provides roadmap from module to chip-on-PCB
- Support for crystal or external clock
- Specific Production mode

# eUniStone PBA31308/2 SPP-AT

## BlueMoon® Universal Family Extended to Embedded Data Transfer Applications

Block Diagram



### BlueMoon Universal Embedded

- 0,13µm CMOS technology
- Bluetooth 2.0 + EDR
- Integrated Stack
- Integrated RFCOMM profile
- AT command interface available
- On-module E²PROM
- On-module 26MHz crystal
- On-module balun/filter
- On-module voltage regulator
- 50Ω Antenna interface

### Interfaces

- High speed UART – 3.25Mbps
- GPIOs with wake-up capability
- Single voltage power supply
- Support direct connect to 2xAAA
- 50Ω balanced Antenna interface
- 26MHz crystal or 26MHz reference frequency
- Optional 32kHz Low Power clock



### Development Kit

The Development Kit is a complete platform for development and evaluation.

- SW for standard applications
- AT command Interface tool
- HCI command Interface tool
- 2 HW boards
- UART interface
- USB to UART converter
- Reference design
- Documentation

### Product Summary

Type	Description	Temperatur Range	Package
eUniStone	PBA31308/2 V1.01	-40C to -70C	Module 8.7 x 11.6mm
	PBA31308/2 V1.11	-40C to +85C	54 pads at 1.20mm pitch

The Bluetooth word mark is owned by the Bluetooth SIG, Inc. and any use of this mark by Infineon Technologies is under license. The BlueMoon trade mark is owned by Infineon Technologies AG.

How to reach us:  
<http://www.infineon.com>

Published by  
Infineon Technologies AG  
81726 Munich, Germany

© 2008 Infineon Technologies AG  
All Rights Reserved.

**Legal Disclaimer** The information given in this Product Brief shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

**Information** For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office ([www.infineon.com](http://www.infineon.com)).

**Warnings** Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office. Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

## B.2 Microcontrolador Infineon XC864 8-bit Single Chip



# XC864

## New 4KB Flash 8-bit Microcontroller Series

### Extends XC800 8-bit MCU Family

XC864 series expands Infineon's 8-bit XC800 microcontroller (MCU) family to provide advanced peripheral integration for motor control performance in a small footprint package (TSSOP-20).

XC864 combines a high performance 8051 core with on-chip Flash memory and powerful on-chip peripherals. In addition, further performance and cost-saving enhancements to the XC800 family include features such as an on-chip oscillator and embedded voltage regulator (EVR), supporting single voltage supply of 3.3V or 5.0V.

#### Applications

- DC Fans, stepper motors
- HID lighting
- Small appliances, power tools
- Battery charging, intelligent sensors, SMPS
- Fail safe controller

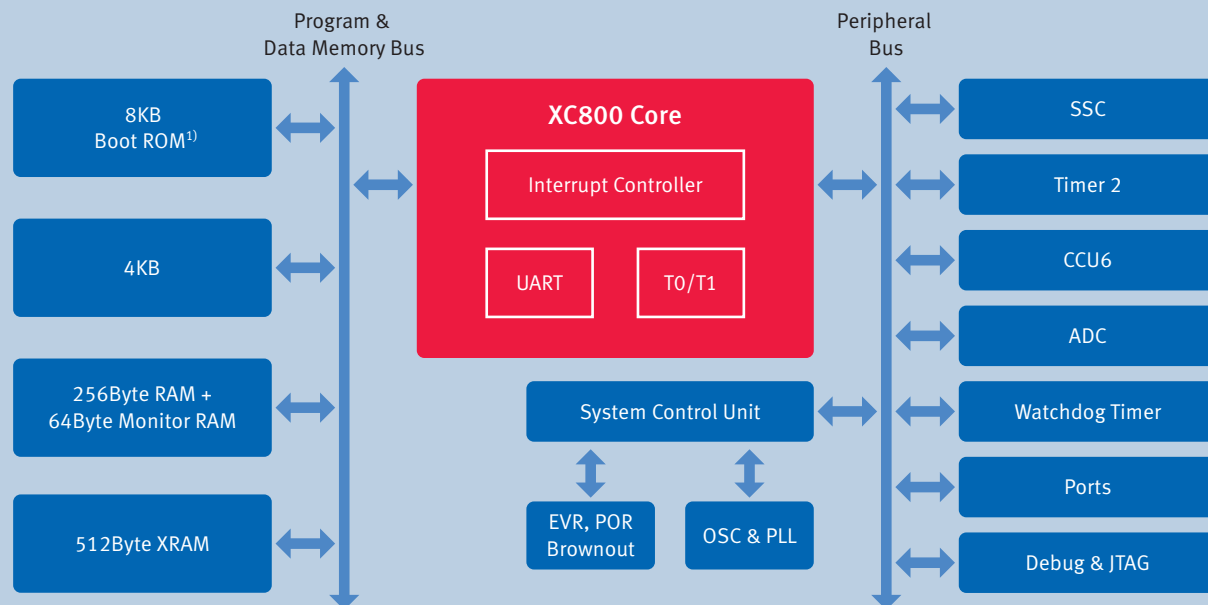
#### Key Features

- High performance XC800 core, compatible to the industry standard 8051 architecture
- 75ns machine cycle time @ 26.67MHz CPU clock
- 2 data pointers
- 4KB of Flash memory Built-in Error Correction
- EEPROM emulation
- 256Byte RAM, 512Byte XRAM
- Capture/Compare Unit (CCU6) dedicated for flexible PWM signal generation for any kind of motor control
  - 3-phase PWM generation
  - 16-bit resolution
  - Support for dead-time generation
- 10-bit ADC with high accuracy (4-channels)
  - Fast conversion in 1.5µs
  - Auto-scan, injection and comparator modes to reduce CPU load
  - Hardware synchronisation to PWM units
- High-Speed SPI Compatible Synchronous Serial Interface (SSC)
- Integrated safety features
  - Brown-out detection
  - Power-on reset
  - Voltage and clock supervisory
  - Window watchdog timer
  - I/O protection circuitry
- On-chip clock generation unit with high accuracy
- 3 general purpose 16-bit timers
- 14 interrupt vectors with 4 priority levels
- Multiple power saving modes available
- On-chip debug support
- JTAG based non-intrusive debugging
- Flexible single voltage supply of 3.3V or 5.0V
- 13 General purpose I/O ports
- Package: TSSOP-20 (green)
  - Small body size : 6.5 x 4.4mm<sup>2</sup>
  - 0.65mm pitch
- Industrial specification
- Temperature range:
  - SAF (-40°C to 85°C)
  - SAK (-40°C to 125°C)

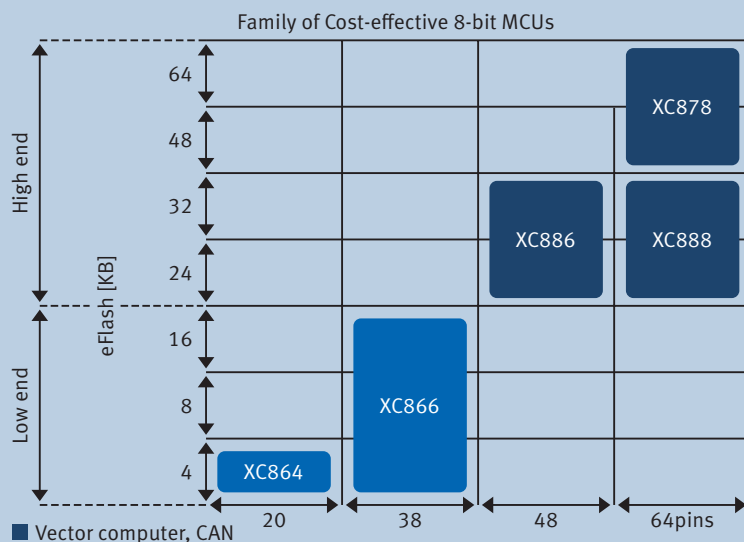
# XC864

## New 4KB Flash 8-bit Microcontroller Series Extends XC800 8-bit MCU Family

Block Diagram



XC800 Family



Key Features

- High performance 8051 core
- 4KB up to 64KB Flash
- EEPROM support
- 20-pin up to 64-pin packages
- Flexible PWM unit
- Fast 10-bit ADC
- On-chip OSC
- LIN, CAN connectivity

How to reach us:  
<http://www.infineon.com>

Published by  
Infineon Technologies AG  
81726 Munich, Germany

© 2008 Infineon Technologies AG  
All Rights Reserved.

**Legal Disclaimer** The information given in this Product Brief shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

**Information** For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office ([www.infineon.com](http://www.infineon.com)).

**Warnings** Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office. Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

### B.3 Bateria EEMB CR1/3N

# CR1/3N

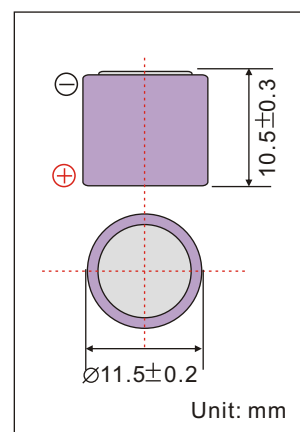
## Brief Datasheet

### Lithium Manganese Dioxide Battery

#### 1 BASIC SPECIFICATION

(Typical values relative to cells stored for one year or less at + 30°C max.)

Nominal capacity ( at 1mA, +23°C) <sup>*1</sup>	160mAh
Nominal voltage	3.0V
Standard discharge current	1mA
Maximum continuous current	5mA
Maximum pulse current <sup>*2</sup>	80mA
Operating temperature range <sup>*3</sup>	-30°C- +60°C
Storage temperature (recommended )	+30°C Max
Weight	3.3g



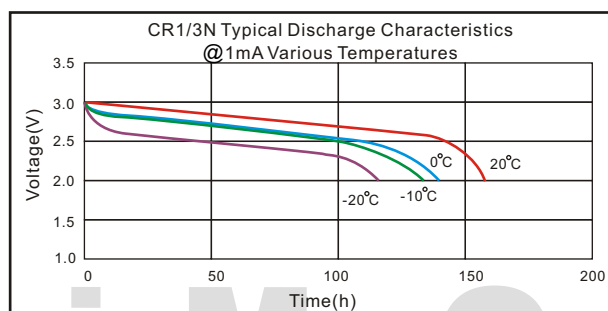
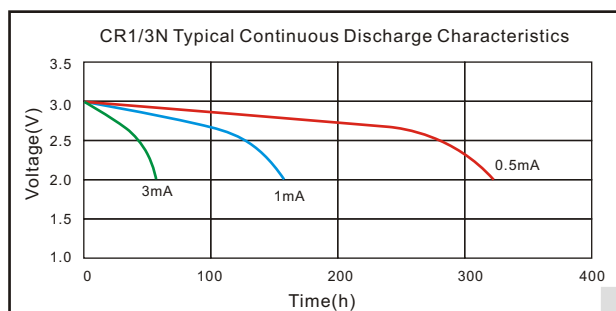
Note:

- \*1 At 1mA discharge current to 2.0V cut-off, +23°C. The capacity restored by the cell varies according to current drain, temperature and cut-off.
- \*2 The readings may vary according to the pulse characteristics, the temperature, and the cell's previous history.
- \*3 Operation above ambient temperature range may lead to reduced capacity and lower voltage readings at the beginning of pulses. Consult EEMB.

#### 2 MAIN FEATURES

- Best suit for high current discharge for both continuous and pulse current.
- Light Weight, High Voltage and High Energy Density.
- Excellent Stable Discharge Characteristics and Outstanding Temperature Characteristics.
- Excellent Leakage Resistance and Excellent Long-term Reliability.

#### 3 ELECTRICAL CHARACTERISTICS



Note: Any representations in this brochure concerning performance, are for informational purposes only and are not construed as warranties either expressed or implied, of future performance.



## **B.4 Comandes/Respostes SPP-AT d’Infineon**

## 4.2 AT Command List Table

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
AT+JAAC=<auto_accept>	<p><b>Auto Accept Connection requests</b> Forces eBMU to accept connection requests.</p> <p><b>auto_accept</b> parameter (1 octet): 0 – Host will be notified on incoming connection indication (+RCOI) – (no auto accept). Default value 1 – eBMU will automatically auto accept incoming connection request – (host will be notified but connection is accepted automatically)</p>	OK	Dev B
AT+JACR= <accept>	<p><b>Accept Connection Request</b> Shall be used as answer to a connect indication (+RCOI).</p> <p><b>accept</b> parameter (1 octet): 0 – Not accepted 1 – Accepted</p>	OK	Dev B
AT+JEDT	<p><b>Enable Device under Test</b> This SPP-AT command enables the device under test. After this command has been sent it is possible for a remote tester to connect, this AT command corresponds to the three different HCI commands listed below:</p> <ol style="list-style-type: none"> <li>1. Set Event Filter – allow all connections.</li> <li>2. Write Scan Enable – page and Inquiry.</li> <li>3. Enable device under test.</li> </ol> <p>Note: <b>JEDT</b> command can only be issued with security mode 1 and after production mode is enabled (JPRO=1). See section 4.8.</p>	OK	Dev A Dev B
AT+JCAC=<trim_value>,<GPIO>	<p><b>Crystal Auto Calibrate</b></p> <p><b>trim_value</b> parameter (4 octets / HEX base): The trim value is used to adjust the frequency on the GPIO chosen by the GPIO parameter.</p> <p><b>GPIO</b> parameter (4 octets / HEX base): Two GPIOs can be used as output for the oscillator trim, P01 and P08 (0002 and 0100). See section 4.4.</p> <p>Note: <b>JCAC</b> command can only be issued after production mode is enabled (JPRO=1). See section 4.8.</p>	<p>+RCACCNF=&lt;osc_trim&gt; <b>osc_trim</b> parameter(4 octets / HEX base)</p> <p>Use AT+JCBD to store the osc_trim value to EEPROM.</p>	Dev A Dev B
AT+JCBD=<bd_data>	<p><b>Change BD_Data</b></p> <p><b>bd_data</b> parameter (116 octets / HEX base): (reference values) msg.bdAddr (12 octets)= see note1 msg.channelWordOffset (4 octets) =0000; msg.clkConf (2 octets)= 8A; msg.eepromSize (2 octets)=80;</p>	OK	Dev A Dev B

## Infineon SPP-AT Command and Response

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	<p>msg.inputFreq (8 octets)=018CBA80;  msg.ImpFeatures (16 octets)=000019987E0602BF;  msg.lpmConf (2 octets)=40;  msg.lpmDrift (2 octets)=FA;  msg.lpmThreshold (2 octets)=12;  msg.ulpmThreshold (2 octets)=18;  msg.pmuConfig (4 octets)=0080;  msg.rfPselD (8 octets)=06050403;  msg.rfPselConf (2 octets)=44;  msg.rssiMin (2 octets)=0C;  msg.rssiMax (2 octets)=10;  msg.ddcTlConf (2 octets)=02;  msg.uartBaudrate (2 octets)=04;  msg.uartInvert (2 octets)=00;  msg.uartPulls (2 octets)=01;  msg.oscSettle (2 octets)=08;  msg.bbConf (2 octets)=04;  msg.rfConf (2 octets)=04;  msg.txPowerRef0 (2 octets)=F2;  msg.txPowerRef1 (2 octets)=F8;  msg.txPowerRef2 (2 octets)=FE;  msg.txPowerRef3 (2 octets)=04;  msg.oscTrim (4 octets)=see note 2;  msg.threeWireArqTimeout (2 octets)=06;  msg.LMP_Version (2 octets)=00;  msg.Reserved (16 octets)=0000000000000000;</p> <p>Note 1: This value shall be configured with the desired Bluetooth device Address (e.g. FFAA010203BB).</p> <p>Note 2: This value shall be configured with the value found during crystal calibration (see command <a href="#">AT+JCAC</a>).</p> <p>Note 3: <a href="#">JCBD</a> command can only be issued after production mode is enabled (<a href="#">JPRO=1</a>). Changes in BD Data take place after production mode command is disabled (<a href="#">JPRO=0</a>) followed by a SW reset (<a href="#">AT+JRES</a>). See section 4.8.</p>		
<a href="#">AT+JCCR</a> =<bd_addr>,<service channel>	<p><b>Create Connection Request</b>  Instructs eBMU to connect to a remote Bluetooth device (prospective slave).</p> <p><a href="#">bd_addr</a> parameter (12 octets / HEX base):  The Bluetooth address of the remote device</p> <p><a href="#">service_channel</a> parameter (2 octets / DEC base):  1-30  Which service channel to connect to can be received from a Service Discovery (<a href="#">AT+JSDS</a>)</p>	<p>OK  (Followed by):  <a href="#">+RCCRCNF</a>=&lt;<a href="#">MTU_size</a>&gt;,&lt;<a href="#">status</a>&gt;  <a href="#">MTU_size</a> parameter (3 octets / DEC base)  <a href="#">status</a> parameter (1 octet)</p> <p>If maximum number of allowed connections already exists:  ERR=-2.</p>	Dev A
<a href="#">AT+JDDS</a> =<limit_inquiry_result>	<p><b>Device Discovery Start</b>  Causes eBMU to start a Device Discovery (Inquiry and Remote Name Request) of the Bluetooth neighborhood.</p>	<p>OK  (Then, if responses are returned):  <a href="#">+RDDSRES</a>=&lt;<a href="#">bd_addr</a>&gt;,&lt;<a href="#">remote_name</a>&gt;,&lt;<a href="#">COD</a>&gt;  (For each response)</p>	Dev A

Infineon SPP-AT Command and Response

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	<p><b>limit_inquiry_result</b> parameter (1 octet): Values range: 0-1. The maximum number of responses that can be obtained is 8.</p> <p>Note: values in the limit_inquiry_result parameter have no impact in the behavior of the command.</p>	<p><b>bd_addr</b> parameter (12 octets / HEX base). <b>remote_name</b> parameter (variable length): Name of the remote device. Page Timeout is received as a name if the remote name is not found during device discovery <b>COD</b> parameter (6 octets / HEX base). Class of device</p> <p>(Completed by): <b>+RDDSCNF=&lt;status&gt;</b> <b>status</b> parameter (1 octet)</p>	
<b>AT+JDIS=&lt;discoverable&gt;</b>	<p><b>DIScoverable</b> Forces eBMU into Page Scan / Inquiry Scan states indefinitely (note: this makes the device discoverable).</p> <p><b>discoverable</b> parameter (1 octet): 0- No scans enabled. 1- Inquiry Scan enabled. 2- Page Scan enabled. 3- Inquiry &amp; Page Scan enabled.</p> <p>Note: Default value after a HW reset is no scans enabled</p>	<p>OK</p>	<p>Dev B</p> <p>Dev A is discoverable but does not accept incoming connections.</p>
<b>AT+JDOI</b>	<p><b>DOWNload Application Image via UART</b> After "OK" response a binary file with the EEPROM image should be sent.</p> <p>File open and send in binary format after "OK" is received.</p> <p>Note: <b>JDOI</b> command can only be issued after production mode is enabled (<b>JPRO=1</b>). Changes in the EEPROM image are effective after production mode command is disabled (<b>JPRO=0</b>) followed by a HW reset. See section 4.8.</p>	<p>OK after command is sent.</p> <p><b>+RDOICNF</b> after .eep file has been written.</p> <p>Note: eBMU calculates the size of the image that is being downloaded from the file header. The response <b>+RDOICNF</b> is generated when the calculated size is reached.</p>	<p>Dev A Dev B</p>
<b>AT+JGPA=&lt;reserved&gt;, &lt;read&gt;, &lt;set&gt;, &lt;clear&gt;</b>	<p><b>GPIO action</b> All parameters of this command are bit fields of 16 bits corresponding to GPIOs P0.15 to P0.0. E.g. if pin P0.0 is the desired bit; the bit field value is 0001 and if the desired bit is P0.12 the bit field value is 0800.</p> <p><b>reserved</b> parameter (4 octets) <b>read</b> parameter (4 octets / Hex base): Values for each bit: 0- No Action 1- Read <b>set</b> parameter (4 octets / HEX base): Values for each bit: 0- No Action 1- Set <b>clear</b> parameter (4 octets / HEX base): Values for each bit:</p>	<p><b>+GPOACNF=&lt;value&gt;</b></p> <p><b>value</b> parameter (4 octets / HEX base): This value is the state of the GPIO PINs specified in read parameter. Values for each bit: 0 – means low. 1 – means high.</p> <p>When the pins are set as outputs the return value will be 0 for the specific pin.</p> <p>Reserved pins will always return 0</p>	<p>Dev A Dev B</p>

Infineon SPP-AT Command and Response

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	0- No Action 1- Clear		
AT+JGPC= <direction>,<open_drain>, <pull_on/off>,<pull_up/down>,<tristate>	<b>GPIO Configuration</b> All parameters of this command are bit fields of 16 bits corresponding to GPIOs P0.15 to P0.0 (See command AT+JGPA).  <b>direction</b> parameter (4 octets / HEX base): Values for each bit: 1 - IN 0 - OUT <b>open_drain</b> parameter (4 octets / HEX base): Values for each bit: 1 - OPEN <b>pull_on/off</b> parameter (4 octets / HEX base): Values for each bit: 1 - Pull ON 0 - Pull OFF <b>pull_up/down</b> parameter (4 octets / HEX base): Value for each bit: 1 - Pull UP <b>tristate</b> parameter (4 octets / HEX base): Value for each bit: 1 - Tri-state  Note: See section 3.2 for available GPIO pins.	OK	Dev A Dev B
AT+JPCR=<length_PIN_code>,<PIN_code>	<b>PIN Code Reply</b> Sent to eBMU in response to a PIN Code Request from a remote Bluetooth device (bd_addr).  <b>length_PIN_code</b> parameter (2 octets / DEC base): Values= 1-16 Length of PIN code <b>PIN_code</b> parameter (length=length_PIN_code): The PIN Code to be sent to the remote Bluetooth device, e.g. AT+JPCR=04,1234  Note: the PIN code is an ASCII string.	OK (Followed by): +RSLE if secure link is established	Dev A Dev B
AT+JPRO=<mode>	<b>PROduction mode</b> <b>mode</b> parameter (1 octet): 1= Production mode ON 0= Production mode OFF  Note: See section 4.8 for commands requiring production mode.	OK	Dev A Dev B
AT+JRBD	<b>Read Bluetooth device Data</b> Sent to eBMU to retrieve the Bluetooth Device Address and the Oscillator trimming value.	+RRBDRES=<bd_addr>,<oscTrim>  <b>bd_addr</b> parameter (12 octets / HEX base): BD Address of remote device  <b>oscTrim</b> parameter (4 octets)	Dev A Dev B

Infineon SPP-AT Command and Response

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
		/ HEX base) Oscillator trim value	
AT+JRES	RESet SW reset of the system.	ROK as applications is restarted.	Dev A Dev B
AT+JRLS=<uuid>, <length_service_name > <service_name>,<service channel >,<CoD>	Register Local Service  uuid parameter (4 octets / HEX base): uuid for supported profile e.g.1101 for Serial Port Profile length_service_name parameter (2 octets / DEC base): Values=1-16 Length of service name service_name parameter (length=length_service_name): Name for the service, no final delimiter is needed service_channel parameter (2 octets / DEC base): Values range: 1-30 Which service channel to connect to; can be received from a Service Discovery with AT+JSDS CoD parameter (6 octets / HEX base) Class of device. The default CoD value is 000000  Note: You may not register more than one service. Note2: This command should be sent if DevB behavior is required.	OK	Dev B
AT+JRRRI	Read Revision Information	+RRRICNF=<revision>  revision parameter (2 octets / HEX base)	Dev A Dev B
AT+JRTD=<bd_addr>	Remove Trusted Device – deletes the trusted device information for a registered device (bd_addr).  bd_addr parameter (12 octets / HEX base): The Bluetooth address of the device that shall be removed from the list	OK	Dev A Dev B
AT+JSCR	Stream Connection Request Connects the SPP and UART streams, transparent communication will be enabled if both sides execute this command.	OK	Dev A Dev B
AT+JSDA=<length>,<data >	Send Data Request  length parameter (3 octets / DEC base): number of bytes to be sent Values range: 1- max MTU data parameter (see note for size): data to be sent  Note: Maximum number of bytes for each packet is reported at connection confirmation (MTU_Size). The value for this parameter is negotiated by the two devices during connection set up.	OK	Dev A Dev B
AT+JSDR	SPP Disconnect Request Forces an SPP disconnection.	OK	Dev A Dev B

## Infineon SPP-AT Command and Response

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
AT+JSDS=<bd_addr>,<uuid>	<p><b>Service Discovery Start</b> Causes eBMU to start a service discovery of device with bd_addr and search for services defined by uuid.</p> <p><b>bd_addr</b> parameter (12 octets / HEX base): BD Address of remote device</p> <p><b>uuid</b> parameter ( 4 octets / HEX base): Service to search for e.g. 1101 for Serial Port Profile</p>	<p>OK (Then, if services are returned): +RSDSRES=&lt;remote_service_name&gt;,&lt;remote_service_channel&gt;</p> <p><b>remote_service_name</b> parameter (variable length): Name of the remote service.</p> <p><b>remote_service_channel</b> parameter (2 octets / DEC base)</p> <p>(For each service) (Completed by): +RSDSCNF=&lt;status&gt; <b>status</b> parameter (1 octet)</p>	Dev A
AT+JSEC=<security_mode>,<Link_key_information>,<PIN_type>,<length_PIN_code>,<PIN_code>	<p><b>Enable SECURITY</b></p> <p><b>security_mode</b> parameter (1 octet):</p> <ol style="list-style-type: none"> <li>Security Mode 1</li> <li>N/A</li> <li>Security Mode 3 (default)</li> </ol> <p><b>link_key_information</b> parameter (1 octet):</p> <ol style="list-style-type: none"> <li>Inform about link key</li> <li>Don't inform about link key (default)</li> </ol> <p><b>PIN_type</b> parameter (1 octet)</p> <ol style="list-style-type: none"> <li>Variable PIN</li> <li>Fixed PIN</li> </ol> <p><b>length_PIN_code</b> parameter (2 octets / DEC base): Length of PIN code. The maximum PIN length value is 16 ( corresponding to a 16 octets long PIN code)</p> <p><b>PIN_code</b> parameter (length=length_PIN_code): Normal user PIN, for example "0000" (default), This parameter is taken into account if PIN_TYPE is fixed.</p> <p>Note: If security mode shall be changed from default settings, AT+JSEC shall be the first command that is sent after a reset. Security mode 3 is the default security mode and the default PIN type is variable.</p>	OK	Dev A Dev B
AT+JSLN=<length_friendly_name>,<friendly_name>	<p><b>Set Local device friendly Name</b> Supports all ASCII characters.</p> <p><b>length_friendly_name</b> parameter (2 octets / DEC base): Length of friendly name, the maximum value for length is 18.</p> <p><b>friendly_name</b> parameter (length=length_friendly_name): No delimiter is required.</p>	OK	Dev A Dev B
AT+JSNF=<sniff_Max>,<sniff_Min>,<sniff_attempt>,<sniff_tmo>,<on/off>	<p><b>Sniff Request</b> Request a link to enter Sniff Mode. All command parameters are given in HEX base with the exception of on/off</p>	<p>+RSNFCNF=&lt;sniff_Interval&gt;,&lt;mode&gt;</p> <p><b>sniff_Interval</b> parameter (4</p>	Dev A Dev B

Infineon SPP-AT Command and Response

AT Command	Usage	Resulting response(s) from eBMU	DevA / Dev B
	<p>parameter. 1 ASCII character represents four bits in HEX base e.g. "A" is 1010. MSB given first.</p> <p><b>sniff_Max</b> parameter (4 octets / HEX base): Maximum allowed sniff interval Value to be written= N Time = N * 0.625 msec Range: 1.25 msec to 40.9 s</p> <p><b>sniff_Min</b> parameter (4 octets / HEX base): Minimum allowed sniff interval Value to be written= N Time = N * 0.625 msec Range: 1.25 msec to 40.9 s</p> <p><b>sniff_attempt</b> parameter (4 octets / HEX base): Number of sniff attempts Value to be written= N Length = N* 1.25 msec Time Range: 0.625msec - 40.9 s</p> <p><b>sniff_tmo</b> parameter (4 octets / HEX base): The time out value for sniff attempts Value to be written= N Time = N * 0.625 msec Range: 0 msec to 40.9 s</p> <p><b>on/off</b> parameter (1 octet): One octet to indicate Sniff ON (value=1) or Sniff OFF (value=0)</p>	<p>octets / HEX base): <b>mode</b> parameter (1 octet): One octet to indicate normal mode (mode=0) or sniff mode (mode=2)</p>	
^^^	<p><b>Stream Connection Cancel</b> End Streaming Mode Send -T0 - ^ -T1- ^ -T1- ^ -T0- with an interval of T1= 1 second between symbols and T0 &gt; 1second.</p> <p>Note: This string is not terminated with CR LF.</p>	OK	Dev A Dev B



### 4.3 AT Responses List (not command triggered)

AT Response	Usage	Command to acknowledge the response	Dev A / Dev B
ROK	Start up response		Dev A Dev B
+RPCI=<bd_addr>	PIN Code Indication  bd_addr parameter (12 octets / HEX base): 1. MSB is sent first.	AT+JPCR=<length_PIN_code>, <PIN_code>  length_PIN_code parameter (2 octets / DEC base): Length of PIN code PIN_code parameter (length=length_PIN_code):	Dev A Dev B
+RCOI=<bd_addr>	Connect Indication  bd_addr parameter (12 octets / HEX base): See previous response for information on the format.	AT+JACR= <accept>  accept parameter (1 octet): 0 – Not accepted 1 – Accepted	Dev B  Note: B should disable connectable mode
+RDAI=<length>,<data>	Data Indication  length parameter (3 octets / DEC base): number of bytes to be sent data parameter (length=length from previous parameter) Received data	Not available during stream mode.	Dev A Dev B
+RDII	Disconnect Indication	Received on the side that has not initiated the disconnection.  During stream mode, if the link is lost, the response +RDDI is received after the link supervision time out which is 20 seconds long.	Dev A Dev B
+RSLE	Secure Link Established		Dev A Dev B
+RSNFCNF=<sniff_interval>,<mode>	Sniff mode confirmation sniff_interval parameter (4 octets / HEX base).  mode parameter (1 octet): One octet to indicate Normal mode (mode=0) or sniff mode (mode=2)	Received on the side that has not issued the sniff command (AT+JSNF)	Dev A Dev A Dev B
+RRSW	Role Switch Indication	Received when the remote device performs a Master/Slave role switch	Dev A Dev B

## B.5 Codi del microcontrolador

Codi B.1: (Microcontrolador) IO.c

```
1 //*****
2 // @Module      GPIO
3 // @Filename    IO.C
4 // @Project     PPST0P.dav
5 //-----
6 // @Controller  Infineon XC864L-1FR
7 // @Compiler    Keil
8 // @Codegenerator 1.1
9 // @Description: This file contains functions that use the IO module.
10 //-----
11 // @Date        25/09/2009 13:38:50
12 //*****
13
14 #include "MAIN.H"
15
16 void IO_vInit(void)
17 {
18
19     /// -----
20     /// Configuration of Port P0:
21     /// -----
22     /// P0.0:
23     /// - is used as general input
24     /// - pull device is disabled (tristate)
25     /// P0.3:
26     /// - is used as general purpose output
27     /// - push/pull output is selected
28     /// - the pin status is high level
29     /// - pull device is disabled (tristate)
30     /// P0.4:
31     /// - is used as general purpose output
32     /// - push/pull output is selected
33     /// - the pin status is hi level    --> on/off
34     /// - pull device is disabled (tristate)
35     /// P0.5:
36     /// - is used as alternate input for the External Interrupt Input 0
37     /// - pull device is disabled (tristate)
38
39     SFR_PAGE(_pp1, noSST);    // switch to page 1
40     P0_PUDSEL = 0xFF;        // load pullup/pulldown select register
41     P0_PUDEN = 0xC4;        // load pullup/pulldown enable register
42
43     SFR_PAGE(_pp0, noSST);    // switch to page 0
44     P0_DIR = 0x18;          // load direction register
45     P0_DATA = 0x18;          // load data output register
46
47     /// -----
48     /// Configuration of Port P1:
49     /// -----
50     /// P1.0:
51     /// - is used as general input
52     /// - pull-up device is assigned
53     /// P1.1:
54     /// - is used as general purpose output
55     /// - push/pull output is selected
56     /// - the pin status is low level
57     /// - pull-up device is assigned
58
59     SFR_PAGE(_pp1, noSST);    // switch to page 1
60     P1_PUDSEL = 0xFF;        // load pullup/pulldown select register
61     P1_PUDEN = 0xFF;        // load pullup/pulldown enable register
62
63     SFR_PAGE(_pp0, noSST);    // switch to page 0
64     P1_DIR = 0x02;          // load direction register
65 }
```

```

66  /// -----
67  /// Configuration of Port P2:
68  /// -----
69  /// P2.7:
70  /// - is used as general input
71  /// - input driver is enabled
72  /// - pull-down device is assigned
73
74  P2_PUDSEL      = 0xFF;          // load pullup/pulldown select register
75  P2_PUDEN       = 0x80;          // load pullup/pulldown enable register
76
77  /// -----
78  /// Configuration of Port P3:
79  /// -----
80  /// P3.0:
81  /// - is used as general purpose output
82  /// - push/pull output is selected
83  /// - the pin status is low level
84  /// - pull device is disabled (tristate)
85  /// P3.1:
86  /// - is used as general purpose output
87  /// - push/pull output is selected
88  /// - the pin status is low level
89  /// - pull device is disabled (tristate)
90
91  P3_PUDSEL      = 0xBF;          // load pullup/pulldown select register
92
93  SFR_PAGE(_pp0, noSST);          // switch to page 0
94  P3_DIR         = 0x03;          // load direction register
95
96 }

```

## Codi B.2: (Microcontrolador) IO.h

```

1  //*****
2  // @Module      GPIO
3  // @Filename     IO.H
4  // @Project     PPSTOP.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler    Keil
8  // @Codegenerator 1.1
9  // @Description: This file contains all function prototypes and macros for
10 //               the IO module.
11 //-----
12 // @Date        25/09/2009 13:38:50
13 //*****
14
15 #ifndef _IO_H_
16 #define _IO_H_
17
18 // Port 0
19 sbit P0_0      = 0x80;
20 sbit P0_1      = 0x81;
21 sbit P0_2      = 0x82;
22 sbit P0_3      = 0x83;
23 sbit P0_4      = 0x84;
24 sbit P0_5      = 0x85;
25
26 // Port 1
27 sbit P1_0      = 0x90;
28 sbit P1_1      = 0x91;
29
30 // Port 2
31 sbit P2_0      = 0xA0;
32 sbit P2_1      = 0xA1;
33 sbit P2_2      = 0xA2;
34 sbit P2_7      = 0xA7;
35

```

```

36 // Port 3
37 sbit P3_0      = 0xB0;
38 sbit P3_1      = 0xB1;
39
40 void IO_vInit(void);
41
42 #endif // ifndef _IO_H_

```

### Codi B.3: (Microcontrolador) MAIN.c

```

1  /*******
2  // @Module      Project Settings
3  // @Filename    MAIN.C
4  // @Project     PPSTOP.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler    Keil
8  // @Codegenerator 1.1
9  // @Description This file contains the Project initialization function.
10 //-----
11 // @Date        25/09/2009 13:38:50
12 /*******
13
14
15 /* Includes */
16 #include "main.h"
17 #include <string.h>
18 #include <stdlib.h>
19
20 /* Defines */
21 #define LF 10      /* Line feed decimal code */
22 #define CR 13      /* Carriage return decimal code */
23 #define MAXDEV 10  /* Max number of devices (OK and KO) stored in memory */
24 #define QUEUE 4    /* Size of circular queue */
25
26 /* States */
27 #define STATE_RESET 1          /* Reset the Infineon Bluetooth device */
28 #define STATE_SETSECURITY 2    /* Set security: no authentication, no encryption */
29 #define STATE_SETNAME 3       /* Set the device name: PPSTOP */
30 #define STATE_SEARCHDEVICES 4  /* Start searching devices */
31 #define STATE_SEARCHDEVICES_ANS 5 /* Analyze answers: found devices */
32 #define STATE_SEARCHSERVICES 6 /* Start service search in found devices */
33 #define STATE_SEARCHSERVICES_ANS 7 /* Analyze answers: services found in current device */
34 /*
35 #define STATE_CHECKSERVICE 8    /* Start service check in PPStop devices */
36 #define STATE_CHECKSERVICE_ANS1 9 /* Analyze answers: connection status */
37 #define STATE_CHECKSERVICE_ANS2 10 /* Analyze answers: OK/KO PPStop service */
38 #define STATE_WAITEVENT 11       /* Wait for humidity sensor event */
39 #define STATE_CONNECTION 12      /* Connect to all PPStop devices */
40 #define STATE_CONNECTION_ANS1 13  /* Analyze answers: connection status */
41 #define STATE_CONNECTION_ANS2 14  /* Analyze answers: info from PPStop device */
42 #define STATE_CHANGESENSIBILITY 15 /* Change humidity sensor sensibility */
43 #define STATE_DISCOVERABLE 16    /* Make device discoverable */
44 #define STATE_ROTATE 17          /* Rotate de devices position */
45
46 char answers[QUEUE][60];          /* Circular queue to read answers from UART */
47 ubyte p_answer_r=0;               /* Circular queue read pointer */
48 ubyte p_answer_w=0;               /* Circular queue write pointer */
49 ubyte p_char=0;                   /* Char pointer in current answer (read) */
50 ubyte queue=0;                    /* Number of available answers */
51 ubyte state=STATE_RESET;          /* Current state */
52
53 /* Function to read data from UART */
54 void readCHAR(){
55     answers[p_answer_w][p_char++]=SBUF; /* Store new char in current slot */
56
57     if (SBUF==LF){
58         answers[p_answer_w][p_char-2]='\0'; /* Carriage Return: end of answer */
59         /* End of answer */

```

```

59         if (p_answer_w==3){                                /* Refresh circular queue write pointer */
60             p_answer_w=0;
61         } else {
62             p_answer_w++;
63         }
64         queue++;                                            /* New answer available */
65         p_char=0;                                          /* Reset char slot pointer */
66     }
67     RI=0;
68 }
69
70 void doReset(){
71     state=STATE_RESET;
72 }
73
74 /* Function to get available answers from circular queue */
75 void readANS(char answer[]){
76     while(!queue);                                        /* Wait for new answers */
77     strcpy(answer,answers[p_answer_r]);                  /* Get new answer */
78     queue--;                                              /* Answer read: slot is free */
79     if (p_answer_r==3){                                  /* Refresh circular queue reader pointer */
80         p_answer_r=0;
81     } else {
82         p_answer_r++;
83     }
84 }
85
86 /* Function to send data to UART */
87 void sendAT(char text[]){
88     unsigned char i;
89     for(i=0;i<strlen(text);i++)                          /* Send all data to UART: char by char */
90         UART_vSendData8(text[i]);
91 }
92
93 /* Function to compare two strings */
94 bit search(char vector[3],char answer[]){
95     return (!strcmp(answer,vector));
96 }
97
98 /* Function to split string */
99 char* split(char answer[], ubyte split){
100     /* Split: {1}={2},{3} */
101     bit copy=0;
102     ubyte i,j=0;
103     char splitstring[13];
104     for(i=0;i<strlen(answer);i++){
105         switch (split){
106             case 1: /* String before '=' */
107                 if (answer[i]!='='){
108                     splitstring[j++]=answer[i];
109                 } else {
110                     i=strlen(answer);
111                 }
112                 break;
113             case 2: /* String between '=' and ',' */
114                 if (answer[i]=='='){
115                     copy=0;
116                     i=strlen(answer);
117                 }
118                 if (copy==1) splitstring[j++]=answer[i];
119                 if (answer[i]==',') copy=1;
120                 break;
121             case 3: /* String after '=' and ',' */
122                 if ((answer[i]==',')&&(copy==1)){
123                     copy=0;
124                     i=strlen(answer);
125                 }
126                 if (copy==1) splitstring[j++]=answer[i];
127                 if (answer[i]==',') copy=1;
128                 break;

```

```

129     }
130 }
131 splitstring[j]='\0';    /* End of string */
132 return splitstring;
133 }
134
135
136 /* Function to change the sensibility of the humidity sensor */
137 void changesensibility(char sen){
138     switch(sen){
139         case 0:    /* 11% */
140             P2_PUDSEL = 0x7F;
141             P3_0 =0;
142             P3_1 =0;
143             P0_3 =1;
144             break;
145         case 1:    /* 33% */
146             P2_PUDSEL = 0x7F;
147             P3_0 =0;
148             P3_1 =1;
149             P0_3 =1;
150             break;
151         default:    /* 11% */
152             P2_PUDSEL = 0x7F;
153             P3_0 =0;
154             P3_1 =0;
155             P0_3 =1;
156             break;
157     }
158 }
159
160 /*
161 Pin TXD_0 (P1.1) is selected for transmission -- groc-blau
162 Pin RXD_0 (P1.0) is selected for reception  -- groc-negre
163 Pin TXD_1 (P0.2) is selected for transmission -- rosa
164 Pin RXD_1 (P0.1) is selected for reception  -- marro(PC)
165 P0.0=UartCTs
166 P0.5=UartRts
167 */
168
169 void MAIN_vInit(void){
170     IO_vInit();                /* Initialization of module 'GPIO' */
171     UART_vInit_bluetooth();    /* Initialization of module 'UART (Serial Interface)' */
172     PS_vInit();                /* Initialization of module 'Power Saving Modes' */
173     IP      = 0x11;            /* Load Interrupt Priority Register */
174     IPH     = 0x10;            /* Load Interrupt Priority High Register */
175     IP1     = 0x00;            /* Load Interrupt Priority 1 Register */
176     IPH1    = 0x00;            /* Load Interrupt Priority 1 High Register */
177     EA      = 1;               /* Globally enable interrupts */
178 }
179
180 void main(void){
181
182     /* Local Variables */
183
184     struct device {
185         char DispOrdre;        /* Device position in list */
186         char DispAddr[13];     /* Device MAC Address */
187         char DispChan[3];      /* Device Connection Channel */
188     } devices[MAXDEV];        /* Current devices */
189
190     ubyte future_state;        /* Future state on connection */
191     ubyte i,j,k,l=0;           /* Counters */
192     ubyte numdevices=0;        /* Number of current devices */
193     bit iniflag=0;             /* Inicialization flag */
194     bit notifyflag=0;          /* Notify flag */
195     bit PPStop_device=0;       /* Boolean for valid PPStop device */
196     char battery[4]="100";     /* Battery status */
197     char sensibility='\0';     /* Sensibility status */
198     char answer[60];           /* Buffer to read answer from UART */

```

```

199
200 /* Main Program */
201
202 MAIN_vInit(); /* Initialization commands */
203 changesensibility(sensibility);
204
205 while(1){
206     switch (state){
207
208         case STATE_RESET: /* Reset the Infineon Bluetooth device */
209             sendAT("AT+JRES\r\n");
210             readANS(answer);
211             if ((search("ROK",answer))){
212                 /* AT command sent successfully */
213                 state++;
214             }
215             break;
216
217         case STATE_SETSECURITY: /* Set security: no authentication, no encryption */
218             sendAT("AT+JSEC=1,1,1,04,1111\r\n");
219             readANS(answer);
220             if ((search("OK",answer))){
221                 /* AT command sent successfully */
222                 state++;
223             } else {
224                 state=STATE_RESET;
225             }
226             break;
227
228         case STATE_SETNAME: /* Set the device name: PPSTOP */
229             sendAT("AT+JSLN=06,PPSTOP\r\n");
230             readANS(&answer);
231             if ((search("OK",answer))){
232                 /* AT command sent successfully */
233                 if (!iniflag){
234                     /* No valid PPStop devices found */
235                     state++;
236                 } else {
237                     /* Valid PPStop devices found: do connection */
238                     i=0;
239                     state=STATE_CONNECTION;
240                 }
241             }
242             break;
243
244         case STATE_SEARCHDEVICES: /* Start searching devices */
245             sendAT("AT+JDDS=0\r\n");
246             readANS(&answer);
247             if ((search("OK",answer))){
248                 /* AT command sent successfully */
249                 state++;
250                 numdevices=0;
251                 i=0;
252                 j=0;
253             }
254             break;
255
256         case STATE_SEARCHDEVICES_ANS: /* Analyze answers: found devices */
257             readANS(&answer);
258             if (search("+RDDSCNF",split(answer,1))){
259                 /* End of devices search */
260                 if (numdevices!=0){
261                     /* Some devices found */
262                     state++;
263                 } else {
264                     /* No devices found */
265                     state=STATE_WAITEVENT;
266                 }
267             } else {
268                 /* Device found: save in memory */
269                 strcpy(devices[numdevices].DispAddr,split(answer,2));
270                 numdevices++;

```

```

269     }
270     break;
271
272     case STATE_SEARCHSERVICES:      /* Start service search in found devices */
273         PPStop_device=0;
274         sendAT("AT+JSDS=");
275         sendAT(devices[i].DispAddr);
276         sendAT(",1101\r\n");
277         readANS(&answer);
278         if ((search("OK",answer))){
279             /* AT command sent successfully */
280             state++;
281         }
282         break;
283
284     case STATE_SEARCHSERVICES_ANS:   /* Analize answers: services found in current
285                                     device */
286         readANS(&answer);
287         if (search("+RSDSCNF",split(answer,1))){
288             /* End of services search in current device */
289             devices[j]=devices[i];
290             i++;
291             if (PPStop_device){
292                 j++;
293             }
294             if (numdevices==i){
295                 /* End of services search in all devices */
296                 numdevices=j;
297                 i=0;
298                 j=0;
299                 state++;
300             } else {
301                 /* Loop for next device */
302                 state--;
303             }
304         } else if (search("PPStop",split(answer,2))){ /* Service found */
305             /* PPStop service found: save in memory */
306             strcpy(devices[i].DispChan,split(answer,3));
307             PPStop_device=1;
308         }
309         break;
310
311     case STATE_CHECKSERVICE:        /* Start service check in PPStop devices */
312         if (numdevices==i){
313             /* End of services check in all PPStop devices */
314             numdevices=j;
315             i=0;
316             j=0;
317             state=STATE_WAITEVENT;
318             TR2 = 0; //para el timer;
319         } else {
320             /* Check service of current PPStop device */
321             PPStop_device=0;
322             sendAT("AT+JCCR=");
323             sendAT(devices[i].DispAddr);
324             sendAT(",");
325             sendAT(devices[i].DispChan);
326             sendAT("\r\n");
327             readANS(&answer);
328             if ((search("OK",answer))){
329                 /* AT command sent successfully */
330                 state++;
331             }
332         }
333         break;
334
335     case STATE_CHECKSERVICE_ANS1:   /* Analize answers: connection status */
336         readANS(&answer);
337         if (search("+RCCRCNF",split(answer,1))){
338             /* Connection status received */

```



```

338         if (search("1",split(answer,3))) {
339             /* Unable to connect: loop for next device */
340             state--;
341             i++;
342         } else {
343             /* Connection successfully established: send check command + status
               */
344             sendAT("AT+JSDA=006,C");
345             sendAT(battery);
346             UART_vSendData8(sensibility);
347             UART_vSendData8(0x31 + j);
348             sendAT("\r\n");
349         }
350     } else if (search("+RDII",answer)) {
351         /* Connection timeout: loop for next device */
352         i++;
353         state--;
354     } else if (search("OK",answer)) {
355         /* AT command sent successfully */
356         state++;
357     }
358     break;
359
360 case STATE_CHECKSERVICE_ANS2: /* Analyze answers: OK/KO PPStop service */
361     readANS(&answer);
362     devices[j]=devices[i];
363     devices[j].DispOrdre=j+1;
364     if (search("+RDAI",split(answer,1))) {
365         /* Data received */
366         if (search("OK",split(answer,3))) {
367             /* PPStop service successfully verified */
368             PPStop_device=1;
369             iniflag=1;
370         }
371     } else if (search("+RDII",answer)) {
372         /* Connection closed */
373         state=STATE_CHECKSERVICE;
374         i++;
375         if (PPStop_device) {
376             j++;
377         }
378     }
379     break;
380
381 case STATE_WAITEVENT: /* Wait for humidity sensor event */
382     /* Switch OFF bluetooth */
383     PO_4 = 0;
384     /* Disable UART */
385     SFR_PAGE(_pp2, noSST); /* switch to page 2 without saving */
386     PO_ALTSEL1 &= ~(ubyte)0x04; /* gpio */
387     SFR_PAGE(_pp0, noSST); /* switch to page 0 without saving */
388     PO_DIR &= ~(ubyte)0x04; /* set input direction */
389     PS_vSetPowerDown(); /* 0x20 = p0.5 alt2(10) == extin0_0 // 00 =
       gpio */
390     /* Enable UART */
391     SFR_PAGE(_pp2, noSST); /* switch to page 2 without saving */
392     PO_ALTSEL1 |= (ubyte)0x04; /* configure alternate function register 1
       */
393     SFR_PAGE(_pp0, noSST); /* switch to page 0 without saving */
394     PO_DIR |= (ubyte)0x04; /* set output direction */
395     /* Switch ON bluetooth */
396     PO_4 = 1;
397     notifyflag=0;
398     break;
399
400 case STATE_CONNECTION: /* Connect to all PPStop devices */
401     if (numdevices==i) {
402         /* End of all device connections */
403         for (i=0; i<75; i++) {
404             for (j=0; j<75; j++) {

```

```

405         for (k=0;k<75;k++);
406     }
407 }
408 state=STATE_WAITEVENT;
409 } else {
410     /* Connect to current PPStop device */
411     sendAT("AT+JCCR=");
412     sendAT(devices[i].DispAddr);
413     sendAT(",");
414     sendAT(devices[i].DispChan);
415     sendAT("\r\n");
416     readANS(&answer);
417     if ((search("OK",answer))) {
418         /* AT command sent successfully */
419         state++;
420     }
421 }
422 break;
423
424 case STATE_CONNECTION_ANS1: /* Analyze answers: connection status */
425     readANS(&answer);
426     if (search("+RCCRCNF",split(answer,1))) {
427         /* Connection status received */
428         if (search("1",split(answer,3))) {
429             /* Unable to connect: loop for next device */
430             state--;
431             i++;
432         } else {
433             /* Connection successfully established: send alarm command + status */
434             if(notifyflag==0){
435                 sendAT("AT+JSDA=006,A");
436             } else {
437                 sendAT("AT+JSDA=006,I");
438             }
439             sendAT(battery);
440             UART_vSendData8(sensibility);
441             UART_vSendData8(0x30 + devices[i].DispOrdre);
442             sendAT("\r\n");
443         }
444     } else if (search("OK",answer)) {
445         /* AT command sent successfully */
446         state++;
447     }
448     break;
449
450 case STATE_CONNECTION_ANS2: /* Analyze answers: info from PPStop device */
451     readANS(&answer);
452     if (search("+RDAI",split(answer,1))) {
453         /* Data received */
454         if (search("OK",split(answer,3))) {
455             /* PPStop device successfully alarmed */
456             future_state=STATE_CONNECTION;
457             i++;
458         } else if (search("B",split(answer,3))) {
459             /* Received: start new search of devices */
460             future_state=STATE_SEARCHDEVICES;
461             i=0;
462         } else if (split(answer,3)[0]=='S') {
463             /* Received: change the humidity sensor sensibility */
464             sensibility=split(answer,3)[1];
465             future_state=STATE_CHANGESSENSIBILITY;
466             i=0;
467         } else if (split(answer,3)[0]=='R') {
468             /* Received: rotate the devices position */
469             future_state=STATE_ROTATE;
470             i=0;
471         }
472     } else if (search("+RDII",answer)) {
473         /* Connection closed */

```

```

474         state=future_state;
475     }
476     break;
477
478     case STATE_CHANGESENSIBILITY:      /* Change humidity sensor sensibility */
479     /* Accions per canviar la sensibilitat del micro */
480     changesensibility(sensibility);
481     state=STATE_WAITEVENT;
482     break;
483
484     case STATE_ROTATE:                  /* rotate the devices position */
485     if(numdevices==1)
486         break;
487     for(i=numdevices;i>0;i--){c
488     strcpy(devices[i].DispChan,devices[i-1].DispChan);
489     strcpy(devices[i].DispAddr,devices[i-1].DispAddr);
490     }
491     strcpy(devices[0].DispChan,devices[numdevices].DispChan);
492     strcpy(devices[0].DispAddr,devices[numdevices].DispAddr);
493     notifyflag=1;
494     state=STATE_CONNECTION;
495     break;
496
497     case STATE_DISCOVERABLE:           /* Make device discoverable */
498     sendAT("AT+JDIS=3\r\n");
499     readANS(&answer);
500     if ((search("OK",answer))) {
501         /* AT command sent successfully */
502         state=STATE_SETNAME;
503     }
504     break;
505
506     } /* End of Switch */
507     } /* End of While */
508 } /* End of Main */

```

#### Codi B.4: (Microcontrolador) MAIN.h

```

1  //*****
2  // @Module      Project Settings
3  // @Filename    MAIN.H
4  // @Project     PPSTOP.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler    Keil
8  // @Codegenerator 1.1
9  // @Description  This is the include header file for all other modules.
10 //-----
11 // @Date        25/09/2009 13:38:50
12 //*****
13
14 #ifndef _MAIN_H_
15 #define _MAIN_H_
16
17 #define MAIN_vUnlockProtecReg() PASSWD = 0x9B
18 #define MAIN_vlockProtecReg() PASSWD = 0xAB
19
20 #define bool    bit
21 #define ulong  unsigned long
22 #define uword  unsigned int
23 #define ubyte  unsigned char
24
25 #define KEIL
26
27 // SFR byte definitions
28 sfr ACC      = 0xE0;
29 sfr ADC_CHCTR0 = 0xCA;
30 sfr ADC_CHCTR1 = 0xCB;
31 sfr ADC_CHCTR2 = 0xCC;

```

```

32 sfr ADC_CHCTR7 = 0xD3;
33 sfr ADC_CHINCR = 0xCB;
34 sfr ADC_CHINFR = 0xCA;
35 sfr ADC_CHINPR = 0xCD;
36 sfr ADC_CHINSR = 0xCC;
37 sfr ADC_CRCR1 = 0xCA;
38 sfr ADC_CMR1 = 0xCC;
39 sfr ADC_CRPR1 = 0xCB;
40 sfr ADC_ETRCR = 0xCF;
41 sfr ADC_EVINCR = 0xCF;
42 sfr ADC_EVINFR = 0xCE;
43 sfr ADC_EVINPR = 0xD3;
44 sfr ADC_EVINSR = 0xD2;
45 sfr ADC_GLOBCTR = 0xCA;
46 sfr ADC_GLOBSTR = 0xCB;
47 sfr ADC_INPCRO = 0xCE;
48 sfr ADC_LCBR = 0xCD;
49 sfr ADC_PAGE = 0xD1;
50 sfr ADC_PRAR = 0xCC;
51 sfr ADC_QOR0 = 0xCF;
52 sfr ADC_QBUR0 = 0xD2;
53 sfr ADC_QINR0 = 0xD2;
54 sfr ADC_QMR0 = 0xCD;
55 sfr ADC_QSR0 = 0xCE;
56 sfr ADC_RCR0 = 0xCA;
57 sfr ADC_RCR1 = 0xCB;
58 sfr ADC_RCR2 = 0xCC;
59 sfr ADC_RCR3 = 0xCD;
60 sfr ADC_RESR0H = 0xCB;
61 sfr ADC_RESR0L = 0xCA;
62 sfr ADC_RESR1H = 0xCD;
63 sfr ADC_RESR1L = 0xCC;
64 sfr ADC_RESR2H = 0xCF;
65 sfr ADC_RESR2L = 0xCE;
66 sfr ADC_RESR3H = 0xD3;
67 sfr ADC_RESR3L = 0xD2;
68 sfr ADC_RESRA0H = 0xCB;
69 sfr ADC_RESRA0L = 0xCA;
70 sfr ADC_RESRA1H = 0xCD;
71 sfr ADC_RESRA1L = 0xCC;
72 sfr ADC_RESRA2H = 0xCF;
73 sfr ADC_RESRA2L = 0xCE;
74 sfr ADC_RESRA3H = 0xD3;
75 sfr ADC_RESRA3L = 0xD2;
76 sfr ADC_VFCR = 0xCE;
77 sfr B = 0xF0;
78 sfr BCON = 0xBD;
79 sfr BG = 0xBE;
80 sfr CCU6_CC60RH = 0xFB;
81 sfr CCU6_CC60RL = 0xFA;
82 sfr CCU6_CC60SRH = 0xFB;
83 sfr CCU6_CC60SRL = 0xFA;
84 sfr CCU6_CC61RH = 0xFD;
85 sfr CCU6_CC61RL = 0xFC;
86 sfr CCU6_CC61SRH = 0xFD;
87 sfr CCU6_CC61SRL = 0xFC;
88 sfr CCU6_CC62RH = 0xFF;
89 sfr CCU6_CC62RL = 0xFE;
90 sfr CCU6_CC62SRH = 0xFF;
91 sfr CCU6_CC62SRL = 0xFE;
92 sfr CCU6_CC63RH = 0x9B;
93 sfr CCU6_CC63RL = 0x9A;
94 sfr CCU6_CC63SRH = 0x9B;
95 sfr CCU6_CC63SRL = 0x9A;
96 sfr CCU6_CMPMODIFH = 0xA7;
97 sfr CCU6_CMPMODIFL = 0xA6;
98 sfr CCU6_CMPSTATH = 0xFF;
99 sfr CCU6_CMPSTATL = 0xFE;
100 sfr CCU6_IENH = 0x9D;
101 sfr CCU6_IENL = 0x9C;

```

```

102 sfr CCU6_INPH   = 0x9F;
103 sfr CCU6_INPL   = 0x9E;
104 sfr CCU6_ISH    = 0x9D;
105 sfr CCU6_ISL    = 0x9C;
106 sfr CCU6_ISRH   = 0xA5;
107 sfr CCU6_ISRL   = 0xA4;
108 sfr CCU6_ISSH   = 0xA5;
109 sfr CCU6_ISSL   = 0xA4;
110 sfr CCU6_MCMCTR  = 0xA7;
111 sfr CCU6_MCMOUTH = 0x9B;
112 sfr CCU6_MCMOUTL = 0x9A;
113 sfr CCU6_MCMOUTSH = 0x9F;
114 sfr CCU6_MCMOUTSL = 0x9E;
115 sfr CCU6_MODCTRH = 0xFD;
116 sfr CCU6_MODCTRL = 0xFC;
117 sfr CCU6_PAGE   = 0xA3;
118 sfr CCU6_PISEL0H = 0x9F;
119 sfr CCU6_PISEL0L = 0x9E;
120 sfr CCU6_PISEL2  = 0xA4;
121 sfr CCU6_PSLR   = 0xA6;
122 sfr CCU6_T12DTCH = 0xA5;
123 sfr CCU6_T12DTCL = 0xA4;
124 sfr CCU6_T12H    = 0xFB;
125 sfr CCU6_T12L    = 0xFA;
126 sfr CCU6_T12MSELH = 0x9B;
127 sfr CCU6_T12MSELL = 0x9A;
128 sfr CCU6_T12PRH  = 0x9D;
129 sfr CCU6_T12PRL  = 0x9C;
130 sfr CCU6_T13H    = 0xFD;
131 sfr CCU6_T13L    = 0xFC;
132 sfr CCU6_T13PRH  = 0x9F;
133 sfr CCU6_T13PRL  = 0x9E;
134 sfr CCU6_TCTROH  = 0xA7;
135 sfr CCU6_TCTR0L  = 0xA6;
136 sfr CCU6_TCTR2H  = 0xFB;
137 sfr CCU6_TCTR2L  = 0xFA;
138 sfr CCU6_TCTR4H  = 0x9D;
139 sfr CCU6_TCTR4L  = 0x9C;
140 sfr CCU6_TRPCTRH = 0xFF;
141 sfr CCU6_TRPCTRL = 0xFE;
142 sfr CMCON        = 0xBA;
143 sfr COCON         = 0xBE;
144 sfr DPH           = 0x83;
145 sfr DPL           = 0x82;
146 sfr EO            = 0xA2;
147 sfr EXICONO       = 0xB7;
148 sfr FDCON         = 0xE9;
149 sfr FDRES         = 0xEB;
150 sfr FDSTEP        = 0xEA;
151 sfr FEAH          = 0xBD;
152 sfr FEAL          = 0xBC;
153 sfr HWBPDR        = 0xF7;
154 sfr HWBPSR        = 0xF6;
155 sfr ID             = 0xB3;
156 sfr IEN0          = 0xA8;
157 sfr IEN1          = 0xE8;
158 sfr IP             = 0xB8;
159 sfr IP1           = 0xF8;
160 sfr IPH           = 0xB9;
161 sfr IPH1          = 0xF9;
162 sfr IRCON0        = 0xB4;
163 sfr IRCON1        = 0xB5;
164 sfr IRCON3        = 0xB4;
165 sfr IRCON4        = 0xB5;
166 sfr MISC_CON      = 0xE9;
167 sfr MMBPCR        = 0xF3;
168 sfr MMCR          = 0xF1;
169 sfr MMCR2         = 0xE9;
170 sfr MMDR          = 0xF5;
171 sfr MMICR         = 0xF4;

```

```

172 sfr MMSR          = 0xF2;
173 sfr MODPISEL      = 0xB3;
174 sfr MODSUSP       = 0xBD;
175 sfr NMICON        = 0xBB;
176 sfr NMISR         = 0xBC;
177 sfr P0_ALTSEL0    = 0x80;
178 sfr P0_ALTSEL1    = 0x86;
179 sfr P0_DATA       = 0x80;
180 sfr P0_DIR        = 0x86;
181 sfr P0_OD         = 0x80;
182 sfr P0_PUDEN      = 0x86;
183 sfr P0_PUDSEL     = 0x80;
184 sfr P1_ALTSEL0    = 0x90;
185 sfr P1_ALTSEL1    = 0x91;
186 sfr P1_DATA       = 0x90;
187 sfr P1_DIR        = 0x91;
188 sfr P1_OD         = 0x90;
189 sfr P1_PUDEN      = 0x91;
190 sfr P1_PUDSEL     = 0x90;
191 sfr P2_DATA       = 0xA0;
192 sfr P2_DIR        = 0xA1;
193 sfr P2_PUDEN      = 0xA1;
194 sfr P2_PUDSEL     = 0xA0;
195 sfr P3_ALTSEL0    = 0xB0;
196 sfr P3_ALTSEL1    = 0xB1;
197 sfr P3_DATA       = 0xB0;
198 sfr P3_DIR        = 0xB1;
199 sfr P3_OD         = 0xB0;
200 sfr P3_PUDEN      = 0xB1;
201 sfr P3_PUDSEL     = 0xB0;
202 sfr PASSWD        = 0xBB;
203 sfr PCON          = 0x87;
204 sfr PLL_CON       = 0xB7;
205 sfr PMCON0        = 0xB4;
206 sfr PMCON1        = 0xB5;
207 sfr PORT_PAGE     = 0xB2;
208 sfr PSW           = 0xD0;
209 sfr SBUF          = 0x99;
210 sfr SCON          = 0x98;
211 sfr SCU_PAGE      = 0xBF;
212 sfr SP            = 0x81;
213 sfr SSC_BRH       = 0xAF;
214 sfr SSC_BRL       = 0xAE;
215 sfr SSC_CONH_0    = 0xAB;
216 sfr SSC_CONH_P    = 0xAB;
217 sfr SSC_CONL_0    = 0xAA;
218 sfr SSC_CONL_P    = 0xAA;
219 sfr SSC_PISEL     = 0xA9;
220 sfr SSC_RBL       = 0xAD;
221 sfr SSC_TBL       = 0xAC;
222 sfr SYSCON0       = 0x8F;
223 sfr T2_RC2H       = 0xC3;
224 sfr T2_RC2L       = 0xC2;
225 sfr T2_T2CON      = 0xC0;
226 sfr T2_T2H        = 0xC5;
227 sfr T2_T2L        = 0xC4;
228 sfr T2_T2MOD      = 0xC1;
229 sfr TCON          = 0x88;
230 sfr TH0           = 0x8C;
231 sfr TH1           = 0x8D;
232 sfr TL0           = 0x8A;
233 sfr TL1           = 0x8B;
234 sfr TMOD          = 0x89;
235 sfr WDTCON        = 0xBB; // located in the mapped SFR area
236 sfr WDTL          = 0xBF; // located in the mapped SFR area
237 sfr WDTL          = 0xBE; // located in the mapped SFR area
238 sfr WDTREL        = 0xBC; // located in the mapped SFR area
239 sfr WDTWINB       = 0xBD; // located in the mapped SFR area
240 sfr XADDRH        = 0xB3;
241

```

```

242 // SFR bit definitions
243
244 // IEN0
245 sbit EA      = 0xAF;
246 sbit ES      = 0xAC;
247 sbit ET0     = 0xA9;
248 sbit ET1     = 0xAB;
249 sbit ET2     = 0xAD;
250 sbit EX0     = 0xA8;
251 sbit EX1     = 0xAA;
252
253 // IEN1
254 sbit EADC    = 0xE8;
255 sbit ECCIP0  = 0xEC;
256 sbit ECCIP1  = 0xED;
257 sbit ECCIP2  = 0xEE;
258 sbit ECCIP3  = 0xEF;
259 sbit ESSC    = 0xE9;
260 sbit EX2     = 0xEA;
261 sbit EXM     = 0xEB;
262
263 // IP1
264 sbit PADC    = 0xF8;
265 sbit PCCIP0  = 0xFC;
266 sbit PCCIP1  = 0xFD;
267 sbit PCCIP2  = 0xFE;
268 sbit PCCIP3  = 0xFF;
269 sbit PSSC    = 0xF9;
270 sbit PX2     = 0xFA;
271 sbit PXM     = 0xFB;
272
273 // IP
274 sbit PS      = 0xBC;
275 sbit PT0     = 0xB9;
276 sbit PT1     = 0xBB;
277 sbit PT2     = 0xBD;
278 sbit PX0     = 0xB8;
279 sbit PX1     = 0xBA;
280
281 // PSW
282 sbit AC      = 0xD6;
283 sbit CY      = 0xD7;
284 sbit F0      = 0xD5;
285 sbit F1      = 0xD1;
286 sbit OV      = 0xD2;
287 sbit P       = 0xD0;
288 sbit RS0     = 0xD3;
289 sbit RS1     = 0xD4;
290
291 // SCON
292 sbit RB8     = 0x9A;
293 sbit REN     = 0x9C;
294 sbit RI      = 0x98;
295 sbit SM0     = 0x9F;
296 sbit SM1     = 0x9E;
297 sbit SM2     = 0x9D;
298 sbit TB8     = 0x9B;
299 sbit TI      = 0x99;
300
301 // T2_T2CON
302 sbit CP_RL2  = 0xC0;
303 sbit EXEN2   = 0xC3;
304 sbit EXF2    = 0xC6;
305 sbit TF2     = 0xC7;
306 sbit TR2     = 0xC2;
307
308 // TCON
309 sbit IE0     = 0x89;
310 sbit IE1     = 0x8B;
311 sbit ITO     = 0x88;

```

```

312 sbit IT1          = 0x8A;
313 sbit TF0          = 0x8D;
314 sbit TF1          = 0x8F;
315 sbit TR0          = 0x8C;
316 sbit TR1          = 0x8E;
317
318
319 // Definition of the 16-bit SFR
320 // sfr16 data type to access two 8-bit SFRs as a single 16-bit SFR.
321
322 sfr16 CCU6_CC60RLH = 0xFA;      // 16-bit Address
323 sfr16 CCU6_CC60SRLH = 0xFA;      // 16-bit Address
324 sfr16 CCU6_CC61RLH = 0xFC;      // 16-bit Address
325 sfr16 CCU6_CC61SRLH = 0xFC;      // 16-bit Address
326 sfr16 CCU6_CC62RLH = 0xFE;      // 16-bit Address
327 sfr16 CCU6_CC62SRLH = 0xFE;      // 16-bit Address
328 sfr16 CCU6_CC63RLH = 0x9A;      // 16-bit Address
329 sfr16 CCU6_CC63SRLH = 0x9A;      // 16-bit Address
330 sfr16 CCU6_T12LH = 0xFA;        // 16-bit Address
331 sfr16 CCU6_T12PRLH = 0x9C;      // 16-bit Address
332 sfr16 CCU6_T13LH = 0xFC;        // 16-bit Address
333 sfr16 CCU6_T13PRLH = 0x9E;      // 16-bit Address
334 sfr16 T2_RC2LH = 0xC2;          // 16-bit Address
335 sfr16 T2_T2LH = 0xC4;           // 16-bit Address
336
337 // Definition of the PAGE SFR
338
339 // PORT_PAGE
340 #define _pp0 PORT_PAGE=0 // PORT_PAGE postfix
341 #define _pp1 PORT_PAGE=1 // PORT_PAGE postfix
342 #define _pp2 PORT_PAGE=2 // PORT_PAGE postfix
343 #define _pp3 PORT_PAGE=3 // PORT_PAGE postfix
344
345 // ADC_PAGE
346 #define _ad0 ADC_PAGE=0 // ADC_PAGE postfix
347 #define _ad1 ADC_PAGE=1 // ADC_PAGE postfix
348 #define _ad2 ADC_PAGE=2 // ADC_PAGE postfix
349 #define _ad3 ADC_PAGE=3 // ADC_PAGE postfix
350 #define _ad4 ADC_PAGE=4 // ADC_PAGE postfix
351 #define _ad5 ADC_PAGE=5 // ADC_PAGE postfix
352 #define _ad6 ADC_PAGE=6 // ADC_PAGE postfix
353
354 // SCU_PAGE
355 #define _su0 SCU_PAGE=0 // SCU_PAGE postfix
356 #define _su1 SCU_PAGE=1 // SCU_PAGE postfix
357 #define _su2 SCU_PAGE=2 // SCU_PAGE postfix
358
359 // CCU6_PAGE
360 #define _cc0 CCU6_PAGE=0 // CCU6_PAGE postfix
361 #define _cc1 CCU6_PAGE=1 // CCU6_PAGE postfix
362 #define _cc2 CCU6_PAGE=2 // CCU6_PAGE postfix
363 #define _cc3 CCU6_PAGE=3 // CCU6_PAGE postfix
364
365 #define SST0 0x80 // Save SFR page to ST0
366 #define RST0 0xC0 // Restore SFR page from ST0
367 #define SST1 0x90 // Save SFR page to ST1
368 #define RST1 0xD0 // Restore SFR page from ST1
369 #define SST2 0xA0 // Save SFR page to ST2
370 #define RST2 0xE0 // Restore SFR page from ST2
371 #define SST3 0xB0 // Save SFR page to ST3
372 #define RST3 0xF0 // Restore SFR page from ST3
373 #define noSST 0x00 // Switch page without saving
374
375 #define SFR_PAGE(pg,op) pg+op
376
377 // SYSCON0_RMAP
378 // The access to the mapped SFR area is enabled.
379 #define SET_RMAP() SYSCON0 |= 0x01
380
381 // The access to the standard SFR area is enabled.

```



```

382 #define RESET_RMAP() SYSCON0 &= ~0x01
383
384
385 #define _su SCU_PAGE // SCU_PAGE
386
387 #define STR_PAGE(pg,op) { _push_(op); \
388     pg ; }
389
390 #define RST_PAGE(op) _pop_(op)
391
392 #include <intrins.h>
393 #include "IO.H"
394 #include "UART.H"
395 #include "T2.H"
396 #include "PS.H"
397 #include <stdio.h>
398
399 #endif // ifndef _MAIN_H_

```

### Codi B.5: (Microcontrolador) PS.c

```

1  //*****
2  // @Module      Power Saving Modes
3  // @Filename    PS.C
4  // @Project     PPST0P.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler    Keil
8  // @Codegenerator 1.1
9  // @Description: This file contains functions that use the PS module.
10 //-----
11 // @Date        25/09/2009 13:38:50
12 //*****
13
14 #include "MAIN.H"
15
16 /* Power initialization */
17 void PS_vInit(void)
18 {
19
20     /// Slow Down Clock is 80,00 MHz
21     /// which is 1/1 of the System Clock fSYS
22     /// wake up without reset is selected
23     SFR_PAGE(_su1, noSST); // switch to page1 with out saving
24     CMCON = 0x00; // load Clock Control Register
25
26     /// Pin P0.5 (input)
27     /// - used as wake up pin via EXINT0 from powerdown
28     PMCON0 = 0x02; // wakeup control register
29
30     // ----- Modules configuration status for PMCON1 reg.-----
31     /// ADC enabled
32     /// SSC enabled
33     /// CCU enabled
34     /// T2 enabled
35     PMCON1 = 0x00; // Power mode control register 1
36     SFR_PAGE(_su0, noSST); // switch to page0 with out saving
37
38     EX0 = 1; // XINTRO is enabled
39
40 } // End of function PS_vInit
41
42 /* Power Down function */
43 void PS_vSetPowerDown(void)
44 {
45
46     /// enter software power down mode
47     STR_PAGE(_su1, _su); // push current SCU page and switch to page1
48

```

```

49  /// External wake-up enabled
50  /// Wake-up via pin EXINT0
51  PMCON0 |= 0x02;
52  MAIN_vUnlockProtecReg();      // open access to protected register
53  PMCON0 |= 0x04;              // set PD
54
55  /// Three nop instructions after PD bit is set to execute first
56  /// instruction correctly after wake-up from powerdown mode
57  _nop_();
58  _nop_();
59  _nop_();
60
61  MAIN_vlockProtecReg();        // close access to protected register
62  RST_PAGE(_su);               // restore the old SCU page
63
64 } // End of function PS_vSetPowerDown

```

### Codi B.6: (Microcontrolador) PS.h

```

1  //*****
2  // @Module      Power Saving Modes
3  // @Filename     PS.H
4  // @Project      PPSTOP.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler     Keil
8  // @Codegenerator 1.1
9  // @Description: This file contains all function prototypes and macros for
10 //               the PS module.
11 //-----
12 // @Date        25/09/2009 13:38:50
13 //*****
14
15 #ifndef _PS_H_
16 #define _PS_H_
17
18 void PS_vInit(void);
19 void PS_vSetPowerDown(void);
20
21 #endif // ifndef _PS_H_

```

### Codi B.7: (Microcontrolador) T2.c

```

1  //*****
2  // @Module      Timer 2
3  // @Filename     T2.C
4  // @Project      PPSTOP.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler     Keil
8  // @Codegenerator 1.1
9  // @Description: This file contains functions that use the T2 module.
10 //-----
11 // @Date        25/09/2009 13:38:50
12 //*****
13
14 #include "MAIN.H"
15
16 ubyte tics=0;
17
18 /* Initialize timer2 */
19 void T2_vInit(void)
20 {
21     T2_RC2LH      = 5;          // 4,8us*5=25us recarrega
22     T2_T2MOD      = 0x1E;       // load timer 2 mode register
23     ET2 = 1;                // Enable interrupt
24 } // End of function T2_vInit
25
26 /* Start timer2 */

```

```

27 void iniTimer()
28 {
29     tics = 0;
30     TR2 = 1;
31 }
32
33 /* Timer2 RAI */
34 void T2_vITmr(void) interrupt T2INT
35 {
36     SFR_PAGE(_su0, SST0);           // switch to page 0
37     if (TF2){
38         // a timer 2 overflow has occurred
39         TF2 = 0;
40         // entra cada 314ms
41         if (++tics>190){
42             TR2 = 0;                // para el timer
43             doReset();              // STATE_RESET
44             tics=0;
45         }
46     }
47     SFR_PAGE(_su0, RST0);           // restore the old page
48 } // End of function T2_vITmr

```

### Codi B.8: (Microcontrolador) T2.h

```

1  //*****
2  // @Module      Timer 2
3  // @Filename    T2.H
4  // @Project     PPSTOP.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler    Keil
8  // @Codegenerator 1.1
9  // @Description: This file contains all function prototypes and macros for
10 //               the T2 module.
11 //-----
12 // @Date        25/09/2009 13:38:50
13 //*****
14
15 #ifndef _T2_H_
16 #define _T2_H_
17
18 void T2_vInit(void);
19 void doReset();
20 void iniTimer();
21 void stopTimer();
22
23 #define T2_vStartTmr() TR2 = 1
24 #define T2_vStopTmr() TR2 = 0
25 #define T2_vSetRC2Reg(Value) T2_RC2LH = Value
26 #define T2INT    5
27
28 #endif // ifndef _T2_H_

```

### Codi B.9: (Microcontrolador) UART.c

```

1  //*****
2  // @Module      UART (Serial Interface)
3  // @Filename    UART.C
4  // @Project     PPSTOP.dav
5  //-----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler    Keil
8  // @Codegenerator 1.1
9  // @Description: This file contains functions that use the UART module.
10 //-----
11 // @Date        25/09/2009 13:38:50
12 //*****
13

```

```

14 #include "MAIN.H"
15
16 /* Function to configure UART for transmission to BT */
17 void UART_vInit_bluetooth(void) //prova
18 {
19
20     /// -----
21     /// UART settings
22     /// -----
23     /// Pin TXD_1 (P0.2) is selected for transmission
24     /// Pin RXD_1 (P0.1) is selected for reception
25     /// Receiver enabled
26     /// Mode 1: 8-bit data, 1 start bit, 1 stop bit, variable baud rate
27     /// BRG is selected for baudrate generation
28
29     SFR_PAGE(_pp2, noSST); // switch to page 2 without saving
30     PO_ALTSEL0  &= ~(ubyte)0x04; // configure alternate function register 0
31     PO_ALTSEL1  |= (ubyte)0x04; // configure alternate function register 1
32
33     SFR_PAGE(_pp0, noSST); // switch to page 0 without saving
34     PO_DIR      |= (ubyte)0x04; // set output direction
35
36     MODPISEL    |= (ubyte)0x01; // configure peripheral input select register
37     BCON        = 0x00; // reset baudrate timer/reload register
38     SCON        = 0x50; // load serial channel control register
39
40     /// -----
41     /// Baudrate generator settings
42     /// -----
43     /// input clock = fPCLK
44     /// Fractional divider is enabled
45     /// baudrate = 115,1843 kbaud
46
47     FDSTEP      = 0xE6; // load fractional divider reload register
48     BG          = 0x0C; // load baudrate timer/reload register
49     FDCON       |= 0x01; // load Fractional Divider control register
50     BCON        |= 0x01; // load baud rate control register
51
52     /// UART interrupt enabled
53     ES = 1;
54 }
55
56
57 /* Function to configure UART for transmission */
58 void UART_vInit_rs232(void)
59 {
60
61     /// -----
62     /// UART settings
63     /// -----
64     /// Pin TXD_0 (P1.1) is selected for transmission
65     /// Pin RXD_0 (P1.0) is selected for reception
66     /// Receiver disabled
67     /// Mode 1: 8-bit data, 1 start bit, 1 stop bit, variable baud rate
68     /// Receiver interrupt flag RI will only be activated if a valid stop bit
69     /// was received
70     /// BRG is selected for baudrate generation
71
72     SFR_PAGE(_pp2, noSST); // switch to page 2 without saving
73     P1_ALTSEL0  &= ~(ubyte)0x02; // configure alternate function register 0
74     P1_ALTSEL1  |= (ubyte)0x02; // configure alternate function register 1
75
76     SFR_PAGE(_pp0, noSST); // switch to page 0 without saving
77     P1_DIR      |= (ubyte)0x02; // set output direction
78
79     MODPISEL    &= ~(ubyte)0x01; // configure peripheral input select register //rs232
80     BCON        = 0x00; // reset baudrate timer/reload register
81     SCON        = 0x50; // load serial channel control register
82
83     /// -----

```

```

84  /// Baudrate generator settings
85  /// -----
86  /// input clock = fPCLK
87  /// Fractional divider is enabled
88  /// baudrate = 115,1843 kbaud
89
90  FDSTEP      = 0xE6;          // load fractional divider reload register
91  BG          = 0x0C;          // load baudrate timer/reload register
92  FDCON       |= 0x01;          // load Fractional Divider control register
93  BCON        |= 0x01;          // load baud rate control register
94
95  /// UART interrupt enabled
96  ES = 1;
97
98 }
99
100 /* Read Interrupt and Write Interrupt RAI */
101 void UART_vIsr(void) interrupt UARTINT
102 {
103     SFR_PAGE(_su0, SST0);      // switch to page 0
104     if (RI){                   // new data available on UART (RI=Read Interrupt)
105         readCHAR();             // read new data from UART
106     }
107     SFR_PAGE(_su0, RST0);      // restore the old page
108 }
109
110 /* Function to send new char to UART */
111 void UART_vSendData8(ubyte ubData)
112 {
113     REN = 0;                   // Clear the receiver enable bit since Tx & Rx pins are single
114         bonded
115     SBUF = ubData;             // Write the transmit data byte, this initiates the transmission.
116     while(!TI){};             // Wait till TI flag set
117     TI = 0;                    // Clear the transmitter interrupt flag
118     REN = 1;                   // Set the receiver enable bit since Tx & Rx pins are single bonded
119 }

```

## Codi B.10: (Microcontrolador) UART.h

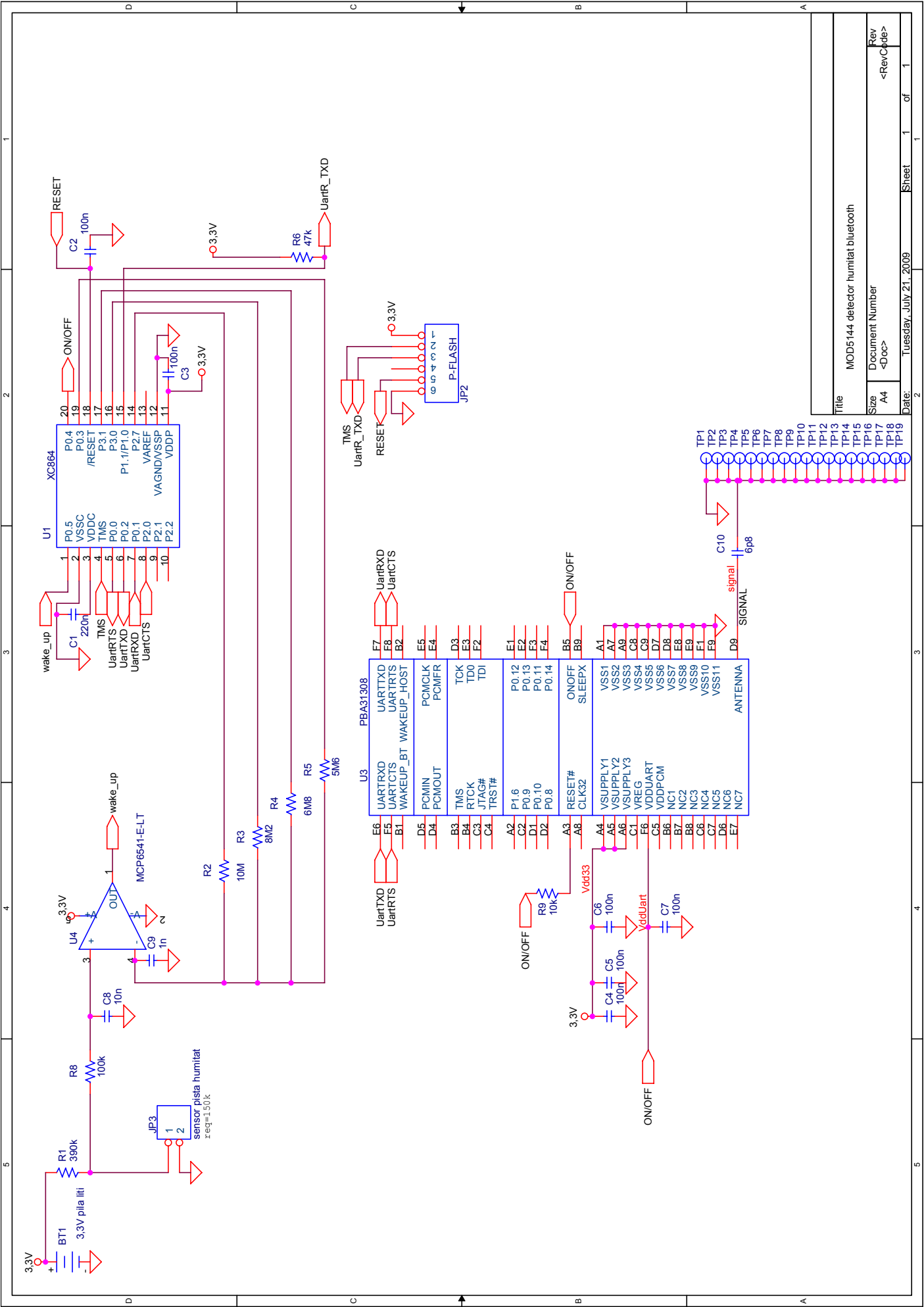
```

1  /*******
2  // @Module      UART (Serial Interface)
3  // @Filename    UART.H
4  // @Project     PPSTOP.dav
5  // -----
6  // @Controller  Infineon XC864L-1FR
7  // @Compiler    Keil
8  // @Codegenerator 1.1
9  // @Description: This file contains all function prototypes and macros for
10 //               the UART module.
11 // -----
12 // @Date        25/09/2009 13:38:50
13 //*****
14
15 #ifndef _UART_H_
16 #define _UART_H_
17
18 void UART_vInit_rs232(void);
19 void UART_vInit_bluetooth(void);
20 void UART_vSendData8(ubyte ubData);
21 void readCHAR();
22
23 #define UART_bTxReady() TI
24 #define UARTINT 4
25
26 #endif // ifndef _UART_H_

```



## B.6 Esquema de connexions



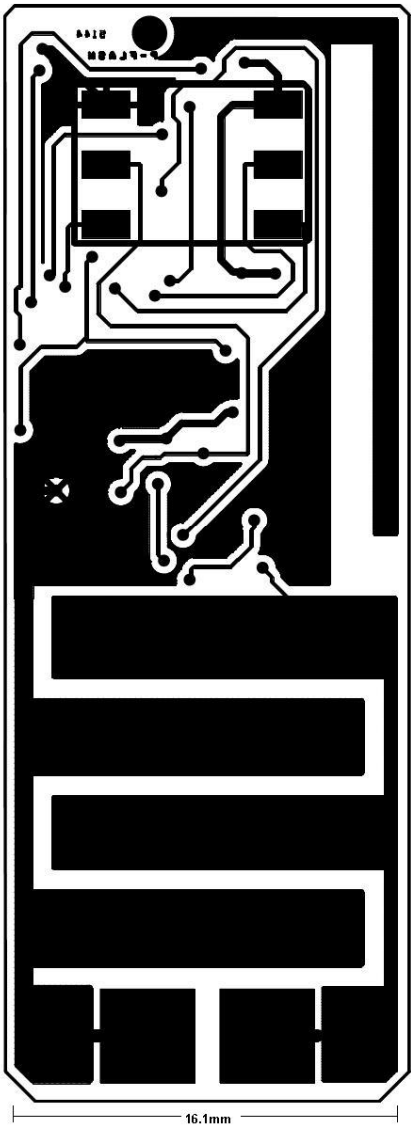
Title		MOD5144 detector humitatat bluetooth	
Size	A4	Document Number	<RevCode>
Rev	<RevCode>		
Date:	Tuesday, July 21, 2009	Sheet	1 of 1



## B.7 Layout, components i encapsulat

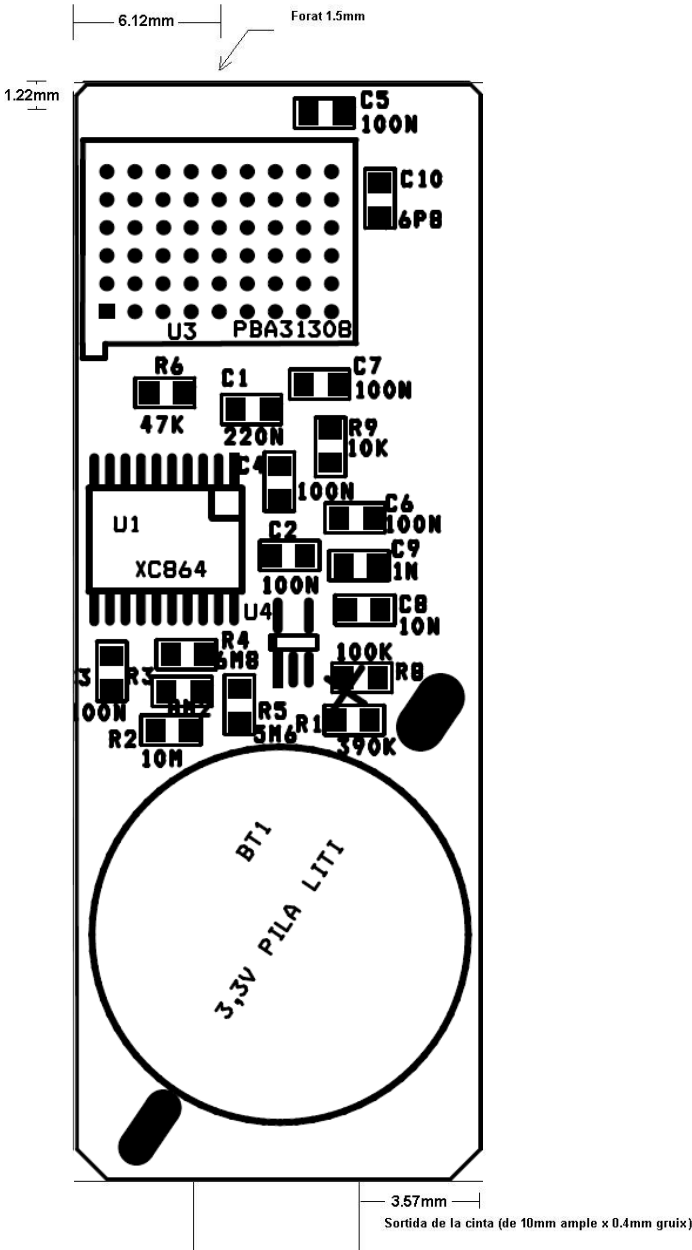
LAYOUT

Cara inferior - versió sense cinta

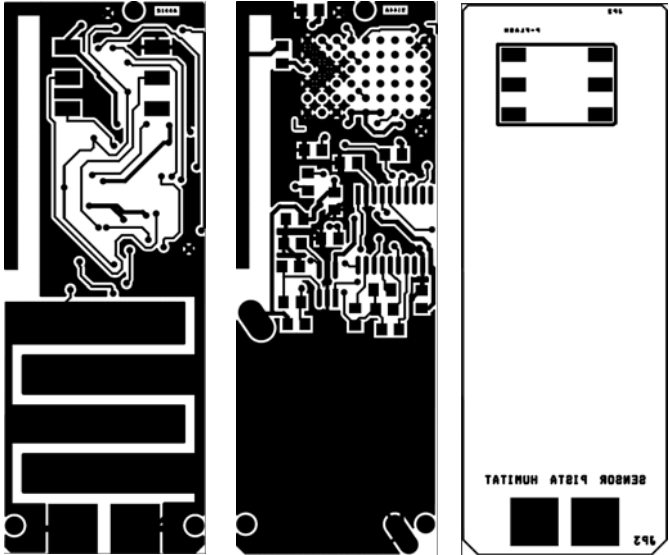
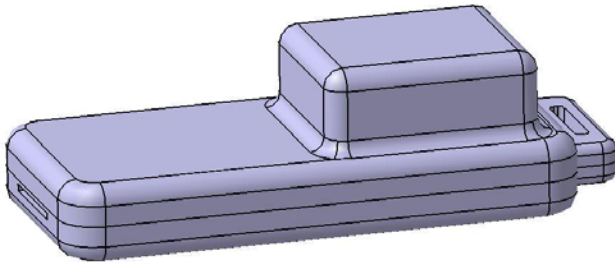


COMPONENTS

Vista cara components



ENCAPSULAT

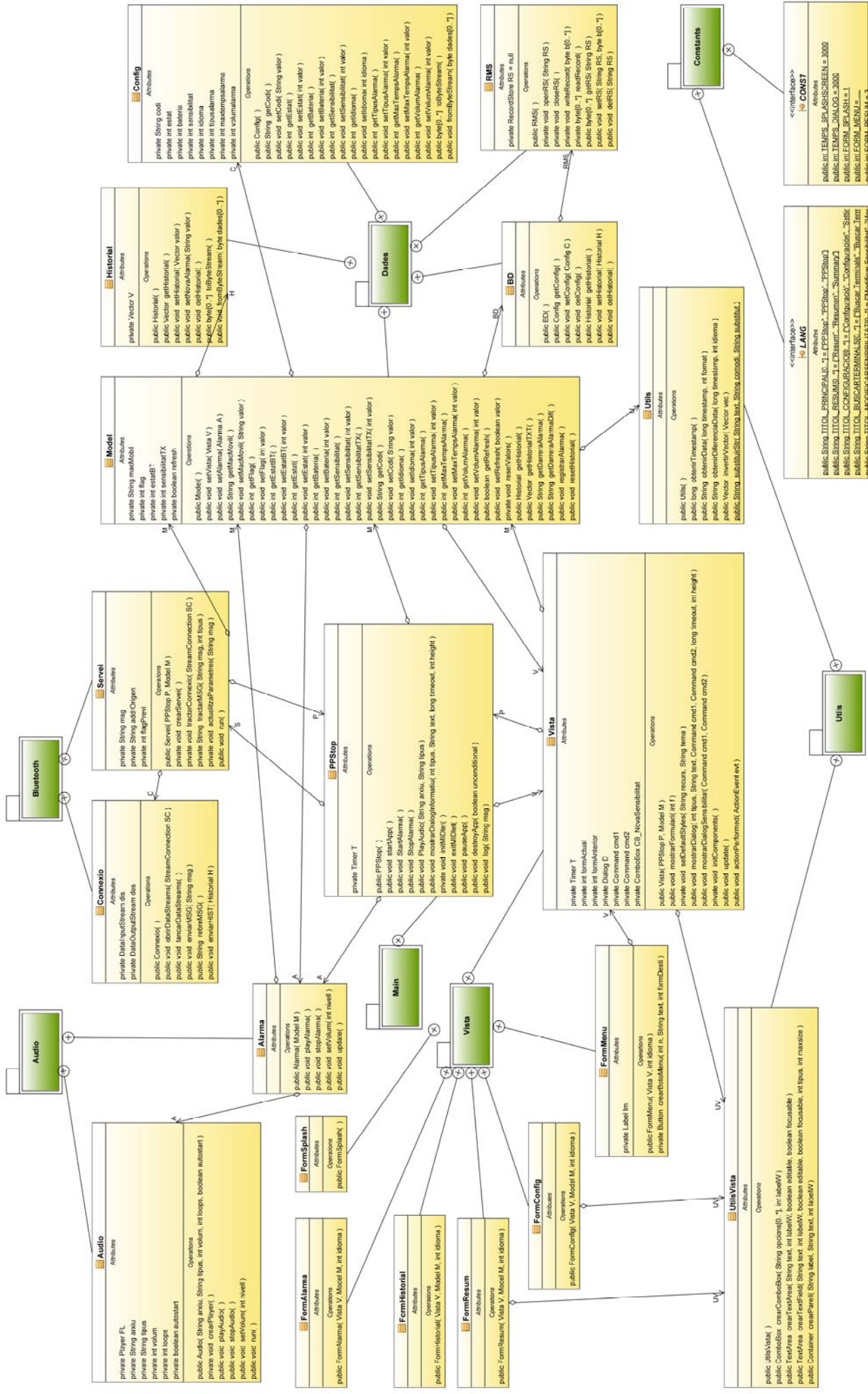


## Apèndix C

### Aplicació de control



## C.1 Diagrama de classes de l'aplicació



## C.2 JavaDoc de l'aplicació

---

# Package Audio

Conté les classes per a la creació, gestió i reproducció d'arxius d'Àudio



# Audio

## Class Alarma

```
java.lang.Object
|
+-Audio.Alarma
```

```
public class Alarma
extends java.lang.Object
```

Classe que crea un player per a la reproducció de l'Alarma PPStop

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

private	<a href="#">A</a> Objecte d'Audio
private	<a href="#">M</a> Referència al Model de dades

### Constructor Summary

public	<a href="#">Alarma</a> ( <a href="#">Model</a> M) Crea una instància de Alarma
--------	---

### Method Summary

void	<a href="#">playAlarma</a> () Activa l'alarma sonora
void	<a href="#">setVolum</a> (int nivell) Ajusta el volum de l'alarma
void	<a href="#">stopAlarma</a> () Atura l'alarma sonora
void	<a href="#">update</a> () Actualitza el player amb la nova informació del Model de Dades

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

(continued on next page)

(continued from last page)

## M

```
private Dades.Model M
```

Referència al Model de dades

## A

```
private Audio.Audio A
```

Objecte d'Audio

## Constructors

### Alarma

```
public Alarma(Model M)
```

Crea una instància de Alarma

**Parameters:**

M - Model de dades

## Methods

### playAlarma

```
public void playAlarma()
```

Activa l'alarma sonora

### stopAlarma

```
public void stopAlarma()
```

Atura l'alarma sonora

### setVolum

```
public void setVolum(int nivell)
```

Ajusta el volum de l'alarma

**Parameters:**

nivell - Nivell de volum (%)

### update

```
public void update()
```

Actualitza el player amb la nova informació del Model de Dades

# Audio

## Class Audio

```

java.lang.Object
  |
  +- java.lang.Thread
        |
        +- Audio.Audio

```

**All Implemented Interfaces:**  
java.lang.Runnable

public class **Audio**  
extends java.lang.Thread

Classe que crea un player per a la reproducció d'arxius d'audio

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

private	<a href="#">arxiu</a> Arxiu a reproduir
private	<a href="#">autostart</a> Començar automàticament l'audio en crear l'objecte
private	<a href="#">loops</a> Nombre de repeticions de reproducció
private	<a href="#">PL</a> Objecte Player
private	<a href="#">tipus</a> Tipus d'arxiu d'audio a reproduir
private	<a href="#">volum</a> Volum de reproducció (%)

### Fields inherited from class java.lang.Thread

blocker, blockerLock, contextClassLoader, daemon, defaultUncaughtExceptionHandler, eetop, EMPTY\_STACK\_TRACE, group, inheritableThreadLocals, inheritedAccessControlContext, MAX\_PRIORITY, MIN\_PRIORITY, name, nativeParkEventPointer, NORM\_PRIORITY, parkBlocker, priority, single\_step, stackSize, stillborn, stopBeforeStart, SUBCLASS\_IMPLEMENTATION\_PERMISSION, subclassAudits, target, threadInitNumber, threadLocals, threadQ, threadSeqNumber, threadStatus, throwableFromStop, tid, uncaughtExceptionHandler

### Constructor Summary

public	<a href="#">Audio</a> (java.lang.String arxiu, java.lang.String tipus, int volum, int loops, boolean autostart) Crea una instància de Audio
--------	--

## Method Summary

void	<a href="#"><code>crearPlayer()</code></a> Crea un player per a la reproducció de l'audio
void	<a href="#"><code>playAudio()</code></a> Inicia la reproducció de l'audio
void	<a href="#"><code>run()</code></a> Crea un player per a la reproducció d'audio i, si escau, el reproduceix
void	<a href="#"><code>setVolum(int nivell)</code></a> Ajusta el volum de l'audio
void	<a href="#"><code>stopAudio()</code></a> Atura la reproducció de l'audio

### Methods inherited from class `java.lang.Thread`

`activeCount`, `auditSubclass`, `blockedOn`, `checkAccess`, `countStackFrames`, `currentThread`, `destroy`, `dispatchUncaughtException`, `dumpStack`, `dumpThreads`, `enumerate`, `exit`, `getAllStackTraces`, `getContextClassLoader`, `getDefaultUncaughtExceptionHandler`, `getId`, `getName`, `getPriority`, `getStackTrace`, `getState`, `getThreadGroup`, `getThreads`, `getUncaughtExceptionHandler`, `holdsLock`, `init`, `interrupt`, `interrupt0`, `interrupted`, `isAlive`, `isCCLOverridden`, `isDaemon`, `isInterrupted`, `isInterrupted`, `join`, `join`, `join`, `nextThreadID`, `nextThreadNum`, `registerNatives`, `resume`, `resume0`, `run`, `setContextClassLoader`, `setDaemon`, `setDefaultUncaughtExceptionHandler`, `setName`, `setPriority`, `setPriority0`, `setUncaughtExceptionHandler`, `sleep`, `sleep`, `start`, `start0`, `stop`, `stop`, `stop0`, `stop1`, `suspend`, `suspend0`, `toString`, `yield`

### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`, `toString`, `wait`, `wait`, `wait`

### Methods inherited from interface `java.lang.Runnable`

`run`

## Fields

### PL

`private javax.microedition.media.Player PL`

Objecte Player

### arxiu

`private java.lang.String arxiu`

Arxiu a reproduir

(continued from last page)

---

## tipus

```
private java.lang.String tipus
```

Tipus d'arxiu d'audio a reproduïr

---

## volum

```
private int volum
```

Volum de reproducció (%)

---

## loops

```
private int loops
```

Nombre de repeticions de reproducció

---

## autostart

```
private boolean autostart
```

Començar automàticament l'audio en crear l'objecte

# Constructors

## Audio

```
public Audio(java.lang.String arxiu,  
             java.lang.String tipus,  
             int volum,  
             int loops,  
             boolean autostart)
```

Crea una instància de Audio

### Parameters:

arxiu - Arxiu a reproduïr  
tipus - Tipus d'arxiu d'audio a reproduïr  
loops - Començar automàticament l'audio en crear l'objecte

# Methods

## crearPlayer

```
private void crearPlayer()
```

Crea un player per a la reproducció de l'audio

---

## playAudio

```
public void playAudio()
```

Inicia la reproducció de l'audio

---

## stopAudio

```
public void stopAudio()
```

(continued from last page)

Atura la reproducció de l'audio

---

## **setVolum**

```
public void setVolum(int nivell)
```

Ajusta el volum de l'audio

### **Parameters:**

nivell - Volum de reproducció (%)

---

## **run**

```
public void run()
```

Crea un player per a la reproducció d'audio i, si escau, el reproduceix

---

## Package

# Bluetooth

Conté les classes per a crear el servei bluetooth i gestionar tota comunicació amb l'aplicació

# Bluetooth

## Class Connexio

```
java.lang.Object
|
+-Bluetooth.Connexio
```

```
public class Connexio
extends java.lang.Object
```

Classe que obre els streams d'e I/O per a la transmissió i recepció de dades

**Author:** Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:** 3.1 (Maig de 2010)

### Field Summary

private	<a href="#">dis</a> Stream d'entrada de dades
private	<a href="#">dos</a> Stream de sortida de dades

### Constructor Summary

public	<a href="#">Connexio()</a> Crea una nova instància de Connexio
--------	---

### Method Summary

void	<a href="#">enviarHIST(<a href="#">Historial</a> H)</a> Transmet l'Historial a través de l'OutputStream
void	<a href="#">enviarMSG(<a href="#">java.lang.String</a> msg)</a> Transmet un Missatge a través de l'OutputStream
void	<a href="#">obrirDataStreams(<a href="#">javax.microedition.io.StreamConnection</a> SC)</a> Obtené els Streams de I/O a partir d'un l'StreamConnection
<a href="#">java.lang.String</a>	<a href="#">rebreMSG()</a> Rep un Missatge a través de l'InputStream
void	<a href="#">tancarDataStreams()</a> Tanca els Streams de I/O

### Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [registerNatives](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Fields



(continued from last page)

## dis

```
private java.io.DataInputStream dis
```

Stream d'entrada de dades

## dos

```
private java.io.DataOutputStream dos
```

Stream de sortida de dades

## Constructors

### Connexio

```
public Connexio()
```

Crea una nova instància de Connexio

## Methods

### obrirDataStreams

```
public void obrirDataStreams(javax.microedition.io.StreamConnection SC)
```

Obtené els Streams de I/O a partir d'un l'StreamConnection

**Parameters:**

SC - Stream de la connexió

### tancarDataStreams

```
public void tancarDataStreams()
```

Tanca els Streams de I/O

### enviarMSG

```
public void enviarMSG(java.lang.String msg)
```

Transmet un Missatge a través de l'OutputStream

**Parameters:**

msg - Missatge a enviar

### rebreMSG

```
public java.lang.String rebreMSG()
```

Rep un Missatge a través de l'InputStream

**Returns:**

Missatge rebut

(continued from last page)

## enviarHIST

```
public void enviarHIST(Historial H)
```

Transmet l'Historial a través de l'OutputStream

### Parameters:

H - Historial a transmetre

# Bluetooth Class Servei

```

java.lang.Object
  |
  +- java.lang.Thread
        +- Bluetooth.Servei

```

**All Implemented Interfaces:**  
java.lang.Runnable

```

public class Servei
extends java.lang.Thread

```

Classe que crea registra el servei bluetooth PPStop i tracta les connexions rebudes

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
3.1 (Maig de 2010)

## Field Summary

private	<a href="#"><u>addrOrigen</u></a> Adreça bluetooth origen de la connexió
private	<a href="#"><u>C</u></a> Objecte de Connexió
private	<a href="#"><u>flagPrevi</u></a> Flag a l'inici de la connexió
private	<a href="#"><u>M</u></a> Referència al Model de dades
private	<a href="#"><u>msg</u></a> Missatge a enviar
private	<a href="#"><u>P</u></a> Referència al MIDlet

## Fields inherited from class java.lang.Thread

blocker, blockerLock, contextClassLoader, daemon, defaultUncaughtExceptionHandler, eetop, EMPTY\_STACK\_TRACE, group, inheritableThreadLocals, inheritedAccessControlContext, MAX\_PRIORITY, MIN\_PRIORITY, name, nativeParkEventPointer, NORM\_PRIORITY, parkBlocker, priority, single\_step, stackSize, stillborn, stopBeforeStart, SUBCLASS\_IMPLEMENTATION\_PERMISSION, subclassAudits, target, threadInitNumber, threadLocals, threadQ, threadSeqNumber, threadStatus, throwableFromStop, tid, uncaughtExceptionHandler

## Constructor Summary

public	<a href="#"><u>Servei</u></a> ( <a href="#"><u>PPStop</u></a> P, <a href="#"><u>Model</u></a> M) Crea una instància de Servei
--------	--

## Method Summary

void	<a href="#"><code>actualitzaParametres</code></a> ( java.lang.String msg) Actualitza en el model la sensibilitat, estat de bateria del dispositiu i ordre del dispositiu
void	<a href="#"><code>crearServei</code></a> () Registra el servei bluetooth PPStop
void	<a href="#"><code>resetParametres</code></a> () Reseteja en el model la sensibilitat, estat de bateria del dispositiu i ordre del dispositiu
void	<a href="#"><code>run</code></a> () Crea el servei bluetooth PPStop i tracta les connexions rebudes
void	<a href="#"><code>tractarConnexio</code></a> ( javax.microedition.io.StreamConnection SC) Tracta la connexió bluetooth rebuda pel servei PPStop
java.lang.String	<a href="#"><code>tractarMSG</code></a> ( java.lang.String msg, int tipus) Tracta un misatge rebut i n'extreu el paràmetre indicat (ORDRE,BATERIA,SENSIBILITAT)

### Methods inherited from class java.lang.Thread

activeCount, auditSubclass, blockedOn, checkAccess, countStackFrames, currentThread, destroy, dispatchUncaughtException, dumpStack, dumpThreads, enumerate, exit, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getThreads, getUncaughtExceptionHandler, holdsLock, init, interrupt, interrupt0, interrupted, isAlive, isCCLOverridden, isDaemon, isInterrupted, isInterrupted, join, join, join, nextThreadID, nextThreadNum, registerNatives, resume, resume0, run, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setPriority0, setUncaughtExceptionHandler, sleep, sleep, start, start0, stop, stop, stop0, stop1, suspend, suspend0, toString, yield

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Methods inherited from interface java.lang.Runnable

run

## Fields

### P

private Main.PPStop **P**

Referència al MIDlet

### M

private Dades.Model **M**

Referència al Model de dades

---

## C

```
private Bluetooth.Connexio C
```

Objecte de Connexió

---

## msg

```
private java.lang.String msg
```

Missatge a enviar

---

## addrOrigen

```
private java.lang.String addrOrigen
```

Adreça bluetooth origen de la connexió

---

## flagPrevi

```
private int flagPrevi
```

Flag a l'inici de la connexió

---

## Constructors

### Servei

```
public Servei(PPStop P,  
              Model M)
```

Crea una instància de Servei

#### Parameters:

P - Referència al MIDlet

M - Referència al Model de dades

## Methods

### crearServei

```
private void crearServei()
```

Registra el servei bluetooth PPStop

---

### tractarConnexio

```
private void tractarConnexio(javax.microedition.io.StreamConnection SC)
```

Tracta la connexió bluetooth rebuda pel servei PPStop

#### Parameters:

SC - Stream de la connexió a tractar

---

(continued from last page)

## tractarMSG

```
private java.lang.String tractarMSG(java.lang.String msg,  
    int tipus)
```

Tracta un missatge rebut i n'extreu el paràmetre indicat (ORDRE,BATERIA,SENSIBILITAT)

### Parameters:

msg - Missatge a tractar

tipus - Tipus de dada a obtenir

### Returns:

Paràmetre sol·licitat

---

## actualitzaParametres

```
private void actualitzaParametres(java.lang.String msg)
```

Actualitza en el model la sensibilitat, estat de bateria del dispositiu i ordre del dispositiu

### Parameters:

msg - Missatge que conté els paràmetres a actualitzar

---

## resetParametres

```
private void resetParametres()
```

Reseteja en el model la sensibilitat, estat de bateria del dispositiu i ordre del dispositiu

---

## run

```
public void run()
```

Crea el servei bluetooth PPStop i tracta les connexions rebudes

---

# Package Constants

Conté les classes de constants numériques i de text

## Constants Interface CONST

public interface **CONST**  
extends

Interfície que conté totes les constants numèriques utilitzades per l'aplicació

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

public static final	<a href="#"><u>ALARMA_ARXIUS</u></a> Arxius d'audio d'alarma: Veu Humana / RING
public static final	<a href="#"><u>ALARMA_RING</u></a> El tipus d'alarma sonora és un RING convencional Value: 1
public static final	<a href="#"><u>ALARMA_SILENCI</u></a> No s'activa cap tipus d'alarma sonora Value: 2
public static final	<a href="#"><u>ALARMA_T10S</u></a> L'alarma es desactiva automàticament passats 10 segons Value: 10000
public static final	<a href="#"><u>ALARMA_T20S</u></a> L'alarma es desactiva automàticament passats 20 segons Value: 20000
public static final	<a href="#"><u>ALARMA_T30S</u></a> L'alarma es desactiva automàticament passats 30 segons Value: 30000
public static final	<a href="#"><u>ALARMA_TINDEFINIT</u></a> L'alarma mai es desactiva automàticament Value: 0
public static final	<a href="#"><u>ALARMA_VEUHUMANA</u></a> El tipus d'alarma sonora és una veu humana Value: 0
public static final	<a href="#"><u>ALARMA_VOL100</u></a> El volum de l'alarma és del 100% Value: 100
public static final	<a href="#"><u>ALARMA_VOL20</u></a> El volum de l'alarma és del 20% Value: 20



public static final	<a href="#"><u>ALARMA_VOL40</u></a> El volum de l'alarma és del 40% Value: <b>40</b>
public static final	<a href="#"><u>ALARMA_VOL60</u></a> El volum de l'alarma és del 60% Value: <b>60</b>
public static final	<a href="#"><u>ALARMA_VOL80</u></a> El volum de l'alarma és del 80% Value: <b>80</b>
public static final	<a href="#"><u>COMMAND_ATURAR</u></a> Command per a aturar l'alarma Value: <b>4</b>
public static final	<a href="#"><u>COMMAND_BUSCARTERMINALS</u></a> Command per a canviar el flag a Buscar Terminals Value: <b>8</b>
public static final	<a href="#"><u>COMMAND_CANCELARACCIO</u></a> Command per a canviar el flag al valor per defecte Value: <b>10</b>
public static final	<a href="#"><u>COMMAND_ESBORRAR</u></a> Command per a esborrar l'història Value: <b>7</b>
public static final	<a href="#"><u>COMMAND_ESBORRARDIALOG</u></a> Command per a confirmació d'esborrar l'història Value: <b>6</b>
public static final	<a href="#"><u>COMMAND_MODIFICARSENSIBILITAT</u></a> Command per a canviar el flag a Modificar Sensibilitat Value: <b>9</b>
public static final	<a href="#"><u>COMMAND_RES</u></a> Command de no dur a terme cap acció Value: <b>1</b>
public static final	<a href="#"><u>COMMAND_ROTARTERMINALS</u></a> Command per a canviar el flag de Rotar Terminals Value: <b>11</b>
public static final	<a href="#"><u>COMMAND_SORTIR</u></a> Command per a sortir de l'aplicació Value: <b>3</b>
public static final	<a href="#"><u>COMMAND_SORTIRDIALOG</u></a> Command per a confirmació de sortir de l'aplicació Value: <b>2</b>
public static final	<a href="#"><u>COMMAND_TORNAR</u></a> Command per a tornar al formulari anterior Value: <b>5</b>

public static final	<a href="#"><u>DATA_CURTA</u></a> Data en format curt Value: 1
public static final	<a href="#"><u>DATA_LLARGA</u></a> Data en format llarg Value: 2
public static final	<a href="#"><u>EMPTY</u></a> Representa un valor buit Value:
public static final	<a href="#"><u>ESTAT_ACTIVAT</u></a> El terminal té la connectivitat bluetooth activada Value: 1
public static final	<a href="#"><u>ESTAT_ASSOCIAT</u></a> El terminal es troba associat a un dispositiu PPStop Value: 1
public static final	<a href="#"><u>ESTAT_DESACTIVAT</u></a> El terminal té la connectivitat bluetooth desactivada Value: 2
public static final	<a href="#"><u>ESTAT_NOASSOCIAT</u></a> El terminal NO es troba associat a cap dispositiu PPStop Value: 2
public static final	<a href="#"><u>FLAG_ALARMA</u></a> La propera acció a realitzar, donada una connexió PPStop, és la d'alarmar Value: 1
public static final	<a href="#"><u>FLAG_BUSCARTERMINALS</u></a> La propera acció a realitzar, donada una connexió PPStop, és la de buscar terminals Value: 2
public static final	<a href="#"><u>FLAG_MODIFICARSENSIBILITAT</u></a> La propera acció a realitzar, donada una connexió PPStop, és la de modificar la sensibilitat Value: 3
public static final	<a href="#"><u>FLAG_ROTARTERMINALS</u></a> La propera acció a realitzar, donada una connexió PPStop, és la de rotar cíclicament la posició dels terminals Value: 4
public static final	<a href="#"><u>FORM_ALARMA</u></a> Formulari de l'Alarma Value: 9
public static final	<a href="#"><u>FORM_ANTERIOR</u></a> Formulari anterior al mostrat actualment Value: 10
public static final	<a href="#"><u>FORM_BUSCARTERMINALS</u></a> Formulari de Buscar Terminals Value: 4

public static final	<a href="#">FORM_CONFIGURACIO</a> Formulari de configuració de l'aplicació Value: 6
public static final	<a href="#">FORM_HISTORIALALARMES</a> Formulari de l'Historial d'Alarmes Value: 7
public static final	<a href="#">FORM_MENU</a> Formulari del Men Principal Value: 2
public static final	<a href="#">FORM_MODIFICARSENSIBILITAT</a> Formulari de Modificar Sensibilitat Value: 5
public static final	<a href="#">FORM_RESUM</a> Formulari de Resum de Dades Value: 3
public static final	<a href="#">FORM_ROTARTERMINALS</a> Formulari de rotació de terminals Value: 11
public static final	<a href="#">FORM_SPLASH</a> Formulari de l'Splash Screen Value: 1
public static final	<a href="#">FORM_TRANSFERIRHISTORIAL</a> Formulari de Transferència de l'Historial Value: 8
public static final	<a href="#">IDIOMA_CA</a> L'idioma de la interfície de l'aplicació és el Català Value: 0
public static final	<a href="#">IDIOMA_EN</a> L'idioma de la interfície de l'aplicació e's l'Anglès Value: 2
public static final	<a href="#">IDIOMA_ES</a> L'idioma de la interfície de l'aplicació és el Castellà Value: 1
public static final	<a href="#">NODEF</a> Representa un valor no definit Value: -1
public static final	<a href="#">OBTENIR_BATERIA</a> Obtenir el paràmetre BATERIA en el MSG rebut Value: 2
public static final	<a href="#">OBTENIR_ORDRE</a> Obtenir el paràmetre ORDRE en el MSG rebut Value: 1

public static final	<a href="#"><u>OBTENIR_ORDRETERMINAL</u></a> Obtenir el paràmetre ORDRE DEL TERMINAL en el MSG rebut Value: <b>4</b>
public static final	<a href="#"><u>OBTENIR_SENSIBILITAT</u></a> Obtenir el paràmetre BATERIA en el MSG rebut Value: <b>3</b>
public static final	<a href="#"><u>ORDRE_ALERTAR</u></a> L'ordre rebuda és la d'alertar (alarma PPStop) Value: <b>A</b>
public static final	<a href="#"><u>ORDRE_COMPROVACIO</u></a> L'ordre rebuda és comprovar codi PPStop: associar servei PPStop Value: <b>C</b>
public static final	<a href="#"><u>ORDRE_INFODISPOSITIU</u></a> L'ordre rebuda és la d'actualitzar la informació del dispositiu PPStop Value: <b>I</b>
public static final	<a href="#"><u>ORDRE_TRANSFERIRHISTORIAL</u></a> L'ordre rebuda és la de transferir historial Value: <b>T</b>
public static final	<a href="#"><u>RS_CONFIG</u></a> RecordSet de la Configuració Value: <b>PPStop_config</b>
public static final	<a href="#"><u>RS_HISTORIAL</u></a> RecordSet de l'Historial Value: <b>PPStop_historial</b>
public static final	<a href="#"><u>SERVEI_NOM</u></a> Nom del servei PPStop a registrar Value: <b>PPStop</b>
public static final	<a href="#"><u>SERVEI_UUID</u></a> UUID del servei PPStop a registrar
public static final	<a href="#"><u>TEMPS_DIALOG</u></a> Temps de Dialog informatiu (ms) Value: <b>3000</b>
public static final	<a href="#"><u>TEMPS_DIALOG_BUSQUEDA</u></a> Temps de Dialog informatiu (ms) de búsqueda en progres... Value: <b>15000</b>
public static final	<a href="#"><u>TEMPS_SPLASHSCREEN</u></a> Temps de l'Splash Screen (ms) Value: <b>3000</b>

## Fields

(continued from last page)

---

## TEMPS\_SPLASHSCREEN

```
public static final int TEMPS_SPLASHSCREEN
```

Temps de l'Splash Screen (ms)  
Constant value: **3000**

---

## TEMPS\_DIALOG

```
public static final int TEMPS_DIALOG
```

Temps de Dialog informatiu (ms)  
Constant value: **3000**

---

## TEMPS\_DIALOG\_BUSQUEDA

```
public static final int TEMPS_DIALOG_BUSQUEDA
```

Temps de Dialog informatiu (ms) de búsqueda en progres...  
Constant value: **15000**

---

## FORM\_SPLASH

```
public static final int FORM_SPLASH
```

Formulari de l'Splash Screen  
Constant value: **1**

---

## FORM\_MENU

```
public static final int FORM_MENU
```

Formulari del Men Principal  
Constant value: **2**

---

## FORM\_RESUM

```
public static final int FORM_RESUM
```

Formulari de Resum de Dades  
Constant value: **3**

---

## FORM\_BUSCARTERMINALS

```
public static final int FORM_BUSCARTERMINALS
```

Formulari de Buscar Terminals  
Constant value: **4**

---

## FORM\_MODIFICARSENSIBILITAT

```
public static final int FORM_MODIFICARSENSIBILITAT
```

Formulari de Modificar Sensibilitat  
Constant value: **5**

---

(continued from last page)

## FORM\_CONFIGURACIO

```
public static final int FORM_CONFIGURACIO
```

Formulari de configuració de l'aplicació  
Constant value: 6

---

## FORM\_HISTORIALALARMES

```
public static final int FORM_HISTORIALALARMES
```

Formulari de l'Historial d'Alarmes  
Constant value: 7

---

## FORM\_TRANSFERIRHISTORIAL

```
public static final int FORM_TRANSFERIRHISTORIAL
```

Formulari de Transferència de l'Historial  
Constant value: 8

---

## FORM\_ALARMA

```
public static final int FORM_ALARMA
```

Formulari de l'Alarma  
Constant value: 9

---

## FORM\_ANTERIOR

```
public static final int FORM_ANTERIOR
```

Formulari anterior al mostrat actualment  
Constant value: 10

---

## FORM\_ROTARTERMINALS

```
public static final int FORM_ROTARTERMINALS
```

Formulari de rotació de terminals  
Constant value: 11

---

## COMMAND\_RES

```
public static final int COMMAND_RES
```

Command de no dur a terme cap acció  
Constant value: 1

---

## COMMAND\_SORTIRDIALOG

```
public static final int COMMAND_SORTIRDIALOG
```

Command per a confirmació de sortir de l'aplicació  
Constant value: 2

---

## COMMAND\_SORTIR

```
public static final int COMMAND_SORTIR
```

(continued from last page)

Command per a sortir de l'aplicació  
Constant value: **3**

---

## COMMAND\_ATURAR

public static final int **COMMAND\_ATURAR**

Command per a aturar l'alarma  
Constant value: **4**

---

## COMMAND\_TORNAR

public static final int **COMMAND\_TORNAR**

Command per a tornar al formulari anterior  
Constant value: **5**

---

## COMMAND\_ESBORRARDIALOG

public static final int **COMMAND\_ESBORRARDIALOG**

Command per a confirmació d'esborrar l'historial  
Constant value: **6**

---

## COMMAND\_ESBORRAR

public static final int **COMMAND\_ESBORRAR**

Command per a esborrar l'historial  
Constant value: **7**

---

## COMMAND\_BUSCARTERMINALS

public static final int **COMMAND\_BUSCARTERMINALS**

Command per a canviar el flag a Buscar Terminals  
Constant value: **8**

---

## COMMAND\_MODIFICARSENSIBILITAT

public static final int **COMMAND\_MODIFICARSENSIBILITAT**

Command per a canviar el flag a Modificar Sensibilitat  
Constant value: **9**

---

## COMMAND\_CANCELARACCIO

public static final int **COMMAND\_CANCELARACCIO**

Command per a canviar el flag al valor per defecte  
Constant value: **10**

---

## COMMAND\_ROTARTERMINALS

public static final int **COMMAND\_ROTARTERMINALS**

Command per a canviar el flag de Rotar Terminals  
Constant value: **11**

---

## NODEF

```
public static final int NODEF
```

Representa un valor no definit  
Constant value: **-1**

---

## ESTAT\_ASSOCIAT

```
public static final int ESTAT_ASSOCIAT
```

El terminal es troba associat a un dispositiu PPStop  
Constant value: **1**

---

## ESTAT\_NOASSOCIAT

```
public static final int ESTAT_NOASSOCIAT
```

El terminal NO es troba associat a cap dispositiu PPStop  
Constant value: **2**

---

## ESTAT\_ACTIVAT

```
public static final int ESTAT_ACTIVAT
```

El terminal té la connectivitat bluetooth activada  
Constant value: **1**

---

## ESTAT\_DESACTIVAT

```
public static final int ESTAT_DESACTIVAT
```

El terminal té la connectivitat bluetooth desactivada  
Constant value: **2**

---

## FLAG\_ALARMA

```
public static final int FLAG_ALARMA
```

La propera acció a realitzar, donada una connexió PPStop, és la d'alarmar  
Constant value: **1**

---

## FLAG\_BUSCARTERMINALS

```
public static final int FLAG_BUSCARTERMINALS
```

La propera acció a realitzar, donada una connexió PPStop, és la de buscar terminals  
Constant value: **2**

---

## FLAG\_MODIFICARSENSIBILITAT

```
public static final int FLAG_MODIFICARSENSIBILITAT
```

La propera acció a realitzar, donada una connexió PPStop, és la de modificar la sensibilitat  
Constant value: **3**

---



(continued from last page)

## FLAG\_ROTARTERMINALS

```
public static final int FLAG_ROTARTERMINALS
```

La propera acció a realitzar, donada una connexió PPStop, és la de rotar cíclicament la posició dels terminals  
Constant value: **4**

---

## IDIOMA\_CA

```
public static final int IDIOMA_CA
```

L'idioma de la interfície de l'aplicació és el Català  
Constant value: **0**

---

## IDIOMA\_ES

```
public static final int IDIOMA_ES
```

L'idioma de la interfície de l'aplicació és el Castellà  
Constant value: **1**

---

## IDIOMA\_EN

```
public static final int IDIOMA_EN
```

L'idioma de la interfície de l'aplicació és l'Anglès  
Constant value: **2**

---

## ALARMA\_VEUHUMANA

```
public static final int ALARMA_VEUHUMANA
```

El tipus d'alarma sonora és una veu humana  
Constant value: **0**

---

## ALARMA\_RING

```
public static final int ALARMA_RING
```

El tipus d'alarma sonora és un RING convencional  
Constant value: **1**

---

## ALARMA\_SILENCI

```
public static final int ALARMA_SILENCI
```

No s'activa cap tipus d'alarma sonora  
Constant value: **2**

---

## ALARMA\_T10S

```
public static final int ALARMA_T10S
```

L'alarma es desactiva automàticament passats 10 segons  
Constant value: **10000**

---

## ALARMA\_T20S

```
public static final int ALARMA_T20S
```

---

(continued from last page)

L'alarma es desactiva automàticament passats 20 segons  
Constant value: **20000**

---

## **ALARMA\_T30S**

```
public static final int ALARMA_T30S
```

L'alarma es desactiva automàticament passats 30 segons  
Constant value: **30000**

---

## **ALARMA\_TINDEFINIT**

```
public static final int ALARMA_TINDEFINIT
```

L'alarma mai es desactiva automàticament  
Constant value: **0**

---

## **ALARMA\_VOL20**

```
public static final int ALARMA_VOL20
```

El volum de l'alarma és del 20%  
Constant value: **20**

---

## **ALARMA\_VOL40**

```
public static final int ALARMA_VOL40
```

El volum de l'alarma és del 40%  
Constant value: **40**

---

## **ALARMA\_VOL60**

```
public static final int ALARMA_VOL60
```

El volum de l'alarma és del 60%  
Constant value: **60**

---

## **ALARMA\_VOL80**

```
public static final int ALARMA_VOL80
```

El volum de l'alarma és del 80%  
Constant value: **80**

---

## **ALARMA\_VOL100**

```
public static final int ALARMA_VOL100
```

El volum de l'alarma és del 100%  
Constant value: **100**

---

## **ALARMA\_ARXIUS**

```
public static final java.lang.String ALARMA_ARXIUS
```

Arxius d'audio d'alarma: Veu Humana / RING

---

(continued from last page)

---

## DATA\_CURTA

```
public static final int DATA_CURTA
```

Data en format curt  
Constant value: **1**

---

## DATA\_LLARGA

```
public static final int DATA_LLARGA
```

Data en format llarg  
Constant value: **2**

---

## OBTENIR\_ORDRE

```
public static final int OBTENIR_ORDRE
```

Obtenir el paràmetre ORDRE en el MSG rebut  
Constant value: **1**

---

## OBTENIR\_BATERIA

```
public static final int OBTENIR_BATERIA
```

Obtenir el paràmetre BATERIA en el MSG rebut  
Constant value: **2**

---

## OBTENIR\_SENSIBILITAT

```
public static final int OBTENIR_SENSIBILITAT
```

Obtenir el paràmetre BATERIA en el MSG rebut  
Constant value: **3**

---

## OBTENIR\_ORDRETERMINAL

```
public static final int OBTENIR_ORDRETERMINAL
```

Obtenir el paràmetre ORDRE DEL TERMINAL en el MSG rebut  
Constant value: **4**

---

## ORDRE\_COMPROVACIO

```
public static final java.lang.String ORDRE_COMPROVACIO
```

L'ordre rebuda és comprovar codi PPStop: associar servei PPStop  
Constant value: **C**

---

## ORDRE\_ALERTAR

```
public static final java.lang.String ORDRE_ALERTAR
```

L'ordre rebuda és la d'alertar (alarma PPStop)  
Constant value: **A**

---

## ORDRE\_TRANSFERIRHISTORIAL

```
public static final java.lang.String ORDRE_TRANSFERIRHISTORIAL
```

(continued from last page)

L'ordre rebuda és la de transferir historial  
Constant value: **T**

---

## ORDRE\_INFODISPOSITIU

public static final java.lang.String **ORDRE\_INFODISPOSITIU**

L'ordre rebuda és la d'actualitzar la informació del dispositiu PPStop  
Constant value: **I**

---

## RS\_CONFIG

public static final java.lang.String **RS\_CONFIG**

RecordSet de la Configuració  
Constant value: **PPStop\_config**

---

## RS\_HISTORIAL

public static final java.lang.String **RS\_HISTORIAL**

RecordSet de l'Historial  
Constant value: **PPStop\_historial**

---

## SERVEI\_NOM

public static final java.lang.String **SERVEI\_NOM**

Nom del servei PPStop a registrar  
Constant value: **PPStop**

---

## SERVEI\_UUID

public static final java.lang.String **SERVEI\_UUID**

UUID del servei PPStop a registrar

---

## EMPTY

public static final java.lang.String **EMPTY**

Representa un valor buit  
Constant value:

## Constants

### Interface LANG

public interface **LANG**  
extends

Interfície que conté totes les constants d'idioma utilitzades per l'aplicació

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

public static final	<a href="#"><u>LABEL_ACTIVAT</u></a> Label del formulari resum: ACTIU
public static final	<a href="#"><u>LABEL_ALTA</u></a> Label del formulari resum: (alta)
public static final	<a href="#"><u>LABEL_ASSOCIAT</u></a> Label del formulari resum: ASSOCIAT
public static final	<a href="#"><u>LABEL_BAIXA</u></a> Label del formulari resum: (baixa)
public static final	<a href="#"><u>LABEL_CA</u></a> Label del formulari configuració: Català
public static final	<a href="#"><u>LABEL_CODIHISTORIAL</u></a> Label del formulari resum: Codi Historial
public static final	<a href="#"><u>LABEL_CODIPPSTOP</u></a> Label del formulari resum: Codi PPStop
public static final	<a href="#"><u>LABEL_CONF_CODIPPSTOP</u></a> Label del formulari configuració: Codi PPStop
public static final	<a href="#"><u>LABEL_CONFIDIOMA</u></a> Label del formulari configuració: Idioma
public static final	<a href="#"><u>LABEL_CONFSENSIBILITAT</u></a> Label del formulari configuració: Sensibilitat
public static final	<a href="#"><u>LABEL_CONFTEMPSALARMA</u></a> Label del formulari configuració: Temps Alarma
public static final	<a href="#"><u>LABEL_CONF_TIPUSALARMA</u></a> Label del formulari configuració: Tipus Alarma
public static final	<a href="#"><u>LABEL_CONFVOLALARMA</u></a> Label del formulari configuració: Volum Alarma
public static final	<a href="#"><u>LABEL_DARRERAALARMA</u></a> Label del formulari resum: Darrera Alarma

public static final	<a href="#"><u>LABEL_DESACTIVAT</u></a> Label del formulari resum: INNACTIU
public static final	<a href="#"><u>LABEL_EN</u></a> Label del formulari configuració: Anglès
public static final	<a href="#"><u>LABEL_ES</u></a> Label del formulari configuració: Castellà
public static final	<a href="#"><u>LABEL_ESTATBATERIA</u></a> Label del formulari resum: Estat Bateria
public static final	<a href="#"><u>LABEL_ESTATBLUETOOTH</u></a> Label del formulari resum: Estat Bluetooth
public static final	<a href="#"><u>LABEL_ESTATSERVEI</u></a> Label del formulari resum: Estat Servei
public static final	<a href="#"><u>LABEL_NOASSOCIAT</u></a> Label del formulari resum: NO ASSOCIAT
public static final	<a href="#"><u>LABEL_ORDRETERMINAL</u></a> Label del formulari resum: Ordre Terminal
public static final	<a href="#"><u>LABEL_PROXIMAACCIO</u></a> Label del formulari resum: Pròxima Acció
public static final	<a href="#"><u>LABEL_RING</u></a> Label del formulari configuració: Ring
public static final	<a href="#"><u>LABEL_SENSIBILITAT</u></a> Label del formulari resum: Sensibilitat
public static final	<a href="#"><u>LABEL_SILENCI</u></a> Label del formulari configuració: Silenci
public static final	<a href="#"><u>LABEL_T10S</u></a> Label del formulari configuració: 10 seg
public static final	<a href="#"><u>LABEL_T20S</u></a> Label del formulari configuració: 20 seg
public static final	<a href="#"><u>LABEL_T30S</u></a> Label del formulari configuració: 30 seg
public static final	<a href="#"><u>LABEL_TINDEFINIT</u></a> Label del formulari configuració: Indefinit
public static final	<a href="#"><u>LABEL_VEUHUMANA</u></a> Label del formulari configuració: Veu Humana
public static final	<a href="#"><u>LABEL_VOL100</u></a> Label del formulari configuració: 100%
public static final	<a href="#"><u>LABEL_VOL20</u></a> Label del formulari configuració: 20%
public static final	<a href="#"><u>LABEL_VOL40</u></a> Label del formulari configuració: 40%

public static final	<a href="#"><u>LABEL_VOL60</u></a> Label del formulari configuració: 60%
public static final	<a href="#"><u>LABEL_VOL80</u></a> Label del formulari configuració: 80%
public static final	<a href="#"><u>MENU_ATURAR</u></a> Label del command d'aturar alarma
public static final	<a href="#"><u>MENU_ESBORRAR</u></a> Label del command d'esborrar historial
public static final	<a href="#"><u>MENU_NO</u></a> Label del command de confirmació NO
public static final	<a href="#"><u>MENU_OK</u></a> Label del command de tancar Dialog Informatiu no finit
public static final	<a href="#"><u>MENU_SI</u></a> Label del command de confirmació SI
public static final	<a href="#"><u>MENU_SORTIR</u></a> Label del command de sortir de l'aplicació
public static final	<a href="#"><u>MENU_TORNAR</u></a> Label del command de tornar al formulari anterior
public static final	<a href="#"><u>TEXT_BUSQUEDAPROGRES</u></a> Text d'informació de búsqueda en progrés
public static final	<a href="#"><u>TEXT_CANCELARACCIO</u></a> Text de confirmació de cancel·lar acció (resetejar flag)
public static final	<a href="#"><u>TEXT_CONFIRMACIOBUSCARTERMINALS</u></a> Text de confirmació per a búsqueda de terminals
public static final	<a href="#"><u>TEXT_CONFIRMACIOMODIFICARSENSIBILITAT</u></a> Text de confirmació per a canvi de sensibilitat
public static final	<a href="#"><u>TEXT_CONFIRMACIOROTARTERMINALS</u></a> Text de confirmació per a rotació de terminals
public static final	<a href="#"><u>TEXT_DISPASSOCIAT</u></a> Text d'informació de dispositiu associat correctament
public static final	<a href="#"><u>TEXT_FA1DIA</u></a> Label del formulari resum: Fa 1 dia
public static final	<a href="#"><u>TEXT_FA1HORA</u></a> Label del formulari resum: Fa 1 hora
public static final	<a href="#"><u>TEXT_FA1MINUT</u></a> Label del formulari resum: Fa 1 minut
public static final	<a href="#"><u>TEXT_FAMENYSDUNMINUT</u></a> Label del formulari resum: Fa < 1 minut
public static final	<a href="#"><u>TEXT_FAXDIES</u></a> Label del formulari resum: Fa {X} dies

public static final	<a href="#"><u>TEXT_FAXHORES</u></a> Label del formulari resum: Fa {X} hores
public static final	<a href="#"><u>TEXT_FAXMINUTS</u></a> Label del formulari resum: Fa {X} mins
public static final	<a href="#"><u>TEXT_FLAGALARMA</u></a> Label del formulari resum: Alarmar
public static final	<a href="#"><u>TEXT_FLAGBUSCARTERMINALS</u></a> Label del formulari resum: Buscar terminals
public static final	<a href="#"><u>TEXT_FLAGCANVIARSENSIBILITAT</u></a> Label del formulari resum: Canvi sen.
public static final	<a href="#"><u>TEXT_FLAGCOMPROVAR</u></a> Label del formulari resum: Comprovar
public static final	<a href="#"><u>TEXT_FLAGROTARTERMINALS</u></a> Label del formulari resum: Rotar terminals
public static final	<a href="#"><u>TEXT_HISTTRANSFERIT</u></a> Text d'informació d'historial d'alarmes transferit correctament
public static final	<a href="#"><u>TEXT_NOALARMES</u></a> Text d'informació d'historial buit
public static final	<a href="#"><u>TEXT_NOASSOCIAT</u></a> Text d'informació de terminal no associat a cap servei PPStop
public static final	<a href="#"><u>TEXT_NOBTACTIU</u></a> Text d'informació de connectivitat bluetooth desactivada
public static final	<a href="#"><u>TEXT_RESETHISTORIAL</u></a> Text de confirmació per a esborrar l'historial
public static final	<a href="#"><u>TEXT_SENSCANVIADA</u></a> Text d'informació de sensibilitat modificada correctament
public static final	<a href="#"><u>TEXT_SORTIR</u></a> Text de confirmació per sortir de l'aplicació
public static final	<a href="#"><u>TEXT_TERMINALROTAT</u></a> Text d'informació de dispositiu associat correctament
public static final	<a href="#"><u>TEXT_TRANSFERIRHISTORIAL</u></a> Text d'informació de transferir historial
public static final	<a href="#"><u>TITOL_ALARMA</u></a> Títol del formulari d'Alarma
public static final	<a href="#"><u>TITOL_BUSCARTERMINALS</u></a> Títol del formulari de Buscar Terminals
public static final	<a href="#"><u>TITOL_CONFIGURACIO</u></a> Títol del formulari de Configuració de l'Aplicació
public static final	<a href="#"><u>TITOL_HISTORIALALARMES</u></a> Títol del formulari d'Historial d'Alarmes



public static final	<a href="#"><u>TITOL_MODIFICARSENSIBILITAT</u></a> Títol del formulari de Modificar Sensibilitat
public static final	<a href="#"><u>TITOL_PRINCIPAL</u></a> Títol del formulari Principal
public static final	<a href="#"><u>TITOL_RESUM</u></a> Títol del formulari Resum de Dades
public static final	<a href="#"><u>TITOL_ROTARTERMINALS</u></a> Títol del formulari de Rotar Terminals
public static final	<a href="#"><u>TITOL_TRANSFERIRHISTORIAL</u></a> Títol del formulari de Transferir Historial

## Fields

### TITOL\_PRINCIPAL

```
public static final java.lang.String TITOL_PRINCIPAL
```

Títol del formulari Principal

### TITOL\_RESUM

```
public static final java.lang.String TITOL_RESUM
```

Títol del formulari Resum de Dades

### TITOL\_CONFIGURACIO

```
public static final java.lang.String TITOL_CONFIGURACIO
```

Títol del formulari de Configuració de l'Aplicació

### TITOL\_BUSCARTERMINALS

```
public static final java.lang.String TITOL_BUSCARTERMINALS
```

Títol del formulari de Buscar Terminals

### TITOL\_MODIFICARSENSIBILITAT

```
public static final java.lang.String TITOL_MODIFICARSENSIBILITAT
```

Títol del formulari de Modificar Sensibilitat

### TITOL\_HISTORIALALARMES

```
public static final java.lang.String TITOL_HISTORIALALARMES
```

Títol del formulari d'Historial d'Alarmes

(continued from last page)

---

## TITOL\_ROTARTERMINALS

```
public static final java.lang.String TITOL_ROTARTERMINALS
```

Títol del formulari de Rotar Terminals

---

## TITOL\_TRANSFERIRHISTORIAL

```
public static final java.lang.String TITOL_TRANSFERIRHISTORIAL
```

Títol del formulari de Transferir Historial

---

## TITOL\_ALARMA

```
public static final java.lang.String TITOL_ALARMA
```

Títol del formulari d'Alarma

---

## MENU\_ATURAR

```
public static final java.lang.String MENU_ATURAR
```

Label del command d'aturar alarma

---

## MENU\_TORNAR

```
public static final java.lang.String MENU_TORNAR
```

Label del command de tornar al formulari anterior

---

## MENU\_SORTIR

```
public static final java.lang.String MENU_SORTIR
```

Label del command de sortir de l'aplicació

---

## MENU\_ESBORRAR

```
public static final java.lang.String MENU_ESBORRAR
```

Label del command d'esborrar historial

---

## MENU\_SI

```
public static final java.lang.String MENU_SI
```

Label del command de confirmació SI

---

## MENU\_NO

```
public static final java.lang.String MENU_NO
```

Label del command de confirmació NO

---

## MENU\_OK

```
public static final java.lang.String MENU_OK
```

---

(continued from last page)

Label del command de tancar Dialog Informatiu no finit

---

## **LABEL\_ESTATBLUETOOTH**

```
public static final java.lang.String LABEL_ESTATBLUETOOTH
```

Label del formulari resum: Estat Bluetooth

---

## **LABEL\_CODIPPSTOP**

```
public static final java.lang.String LABEL_CODIPPSTOP
```

Label del formulari resum: Codi PPStop

---

## **LABEL\_CODIHISTORIAL**

```
public static final java.lang.String LABEL_CODIHISTORIAL
```

Label del formulari resum: Codi Historial

---

## **LABEL\_ESTATSERVEI**

```
public static final java.lang.String LABEL_ESTATSERVEI
```

Label del formulari resum: Estat Servei

---

## **LABEL\_ORDRETERMINAL**

```
public static final java.lang.String LABEL_ORDRETERMINAL
```

Label del formulari resum: Ordre Terminal

---

## **LABEL\_ESTATBATERIA**

```
public static final java.lang.String LABEL_ESTATBATERIA
```

Label del formulari resum: Estat Bateria

---

## **LABEL\_SENSIBILITAT**

```
public static final java.lang.String LABEL_SENSIBILITAT
```

Label del formulari resum: Sensibilitat

---

## **LABEL\_DARRERAALARMA**

```
public static final java.lang.String LABEL_DARRERAALARMA
```

Label del formulari resum: Darrera Alarma

---

## **LABEL\_PROXIMAACCIO**

```
public static final java.lang.String LABEL_PROXIMAACCIO
```

Label del formulari resum: Pròxima Acció

(continued from last page)

---

## **LABEL\_BAIXA**

```
public static final java.lang.String LABEL_BAIXA
```

Label del formulari resum: (baixa)

---

## **LABEL\_ALTA**

```
public static final java.lang.String LABEL_ALTA
```

Label del formulari resum: (alta)

---

## **LABEL\_NOASSOCIAT**

```
public static final java.lang.String LABEL_NOASSOCIAT
```

Label del formulari resum: NO ASSOCIAT

---

## **LABEL\_ASSOCIAT**

```
public static final java.lang.String LABEL_ASSOCIAT
```

Label del formulari resum: ASSOCIAT

---

## **LABEL\_ACTIVAT**

```
public static final java.lang.String LABEL_ACTIVAT
```

Label del formulari resum: ACTIU

---

## **LABEL\_DESACTIVAT**

```
public static final java.lang.String LABEL_DESACTIVAT
```

Label del formulari resum: INNACTIU

---

## **TEXT\_FLAGCOMPROVAR**

```
public static final java.lang.String TEXT_FLAGCOMPROVAR
```

Label del formulari resum: Comprovar

---

## **TEXT\_FLAGALARMA**

```
public static final java.lang.String TEXT_FLAGALARMA
```

Label del formulari resum: Alarmar

---

## **TEXT\_FLAGBUSCARTERMINALS**

```
public static final java.lang.String TEXT_FLAGBUSCARTERMINALS
```

Label del formulari resum: Buscar terminals

---

## **TEXT\_FLAGCANVIARSENSIBILITAT**

```
public static final java.lang.String TEXT_FLAGCANVIARSENSIBILITAT
```

(continued from last page)

Label del formulari resum: Canvi sen.

---

## TEXT\_FLAGROTARTERMINALS

```
public static final java.lang.String TEXT_FLAGROTARTERMINALS
```

Label del formulari resum: Rotar terminals

---

## TEXT\_FAMENYSDUNMINUT

```
public static final java.lang.String TEXT_FAMENYSDUNMINUT
```

Label del formulari resum: Fa < 1 minut

---

## TEXT\_FA1MINUT

```
public static final java.lang.String TEXT_FA1MINUT
```

Label del formulari resum: Fa 1 minut

---

## TEXT\_FAXMINUTS

```
public static final java.lang.String TEXT_FAXMINUTS
```

Label del formulari resum: Fa {X} mins

---

## TEXT\_FA1HORA

```
public static final java.lang.String TEXT_FA1HORA
```

Label del formulari resum: Fa 1 hora

---

## TEXT\_FAXHORES

```
public static final java.lang.String TEXT_FAXHORES
```

Label del formulari resum: Fa {X} hores

---

## TEXT\_FA1DIA

```
public static final java.lang.String TEXT_FA1DIA
```

Label del formulari resum: Fa 1 dia

---

## TEXT\_FAXDIES

```
public static final java.lang.String TEXT_FAXDIES
```

Label del formulari resum: Fa {X} dies

---

## LABEL\_CONFCODIPPSTOP

```
public static final java.lang.String LABEL_CONFCODIPPSTOP
```

Label del formulari configuració: Codi PPStop

---

(continued from last page)

---

## **LABEL\_CONF TIPUSALARMA**

```
public static final java.lang.String LABEL_CONF TIPUSALARMA
```

Label del formulari configuració: Tipus Alarma

---

## **LABEL\_CONF IDIOMA**

```
public static final java.lang.String LABEL_CONF IDIOMA
```

Label del formulari configuració: Idioma

---

## **LABEL\_CONF TEMPSALARMA**

```
public static final java.lang.String LABEL_CONF TEMPSALARMA
```

Label del formulari configuració: Temps Alarma

---

## **LABEL\_CONF VOLALARMA**

```
public static final java.lang.String LABEL_CONF VOLALARMA
```

Label del formulari configuració: Volum Alarma

---

## **LABEL\_CONF SENSIBILITAT**

```
public static final java.lang.String LABEL_CONF SENSIBILITAT
```

Label del formulari configuració: Sensibilitat

---

## **LABEL\_CA**

```
public static final java.lang.String LABEL_CA
```

Label del formulari configuració: Català

---

## **LABEL\_ES**

```
public static final java.lang.String LABEL_ES
```

Label del formulari configuració: Castellà

---

## **LABEL\_EN**

```
public static final java.lang.String LABEL_EN
```

Label del formulari configuració: Anglès

---

## **LABEL\_VE UHUMANA**

```
public static final java.lang.String LABEL_VE UHUMANA
```

Label del formulari configuració: Veu Humana

---

## **LABEL\_RING**

```
public static final java.lang.String LABEL_RING
```

(continued from last page)

Label del formulari configuració: Ring

---

## **LABEL\_SILENCI**

```
public static final java.lang.String LABEL_SILENCI
```

Label del formulari configuració: Silenci

---

## **LABEL\_T10S**

```
public static final java.lang.String LABEL_T10S
```

Label del formulari configuració: 10 seg

---

## **LABEL\_T20S**

```
public static final java.lang.String LABEL_T20S
```

Label del formulari configuració: 20 seg

---

## **LABEL\_T30S**

```
public static final java.lang.String LABEL_T30S
```

Label del formulari configuració: 30 seg

---

## **LABEL\_TINDEFINIT**

```
public static final java.lang.String LABEL_TINDEFINIT
```

Label del formulari configuració: Indefinit

---

## **LABEL\_VOL20**

```
public static final java.lang.String LABEL_VOL20
```

Label del formulari configuració: 20%

---

## **LABEL\_VOL40**

```
public static final java.lang.String LABEL_VOL40
```

Label del formulari configuració: 40%

---

## **LABEL\_VOL60**

```
public static final java.lang.String LABEL_VOL60
```

Label del formulari configuració: 60%

---

## **LABEL\_VOL80**

```
public static final java.lang.String LABEL_VOL80
```

Label del formulari configuració: 80%

---

(continued from last page)

---

## **LABEL\_VOL100**

```
public static final java.lang.String LABEL_VOL100
```

Label del formulari configuració: 100%

---

## **TEXT\_SORTIR**

```
public static final java.lang.String TEXT_SORTIR
```

Text de confirmació per sortir de l'aplicació

---

## **TEXT\_NOASSOCIAT**

```
public static final java.lang.String TEXT_NOASSOCIAT
```

Text d'informació de terminal no associat a cap servei PPStop

---

## **TEXT\_CONFIRMACIOBUSCARTERMINALS**

```
public static final java.lang.String TEXT_CONFIRMACIOBUSCARTERMINALS
```

Text de confirmació per a búsqueda de terminals

---

## **TEXT\_CONFIRMACIOMODIFICARSENSIBILITAT**

```
public static final java.lang.String TEXT_CONFIRMACIOMODIFICARSENSIBILITAT
```

Text de confirmació per a canvi de sensibilitat

---

## **TEXT\_CONFIRMACIOROTARTERMINALS**

```
public static final java.lang.String TEXT_CONFIRMACIOROTARTERMINALS
```

Text de confirmació per a rotació de erminals

---

## **TEXT\_CANCELARACCIO**

```
public static final java.lang.String TEXT_CANCELARACCIO
```

Text de confirmació de cancel·lar acció (resetejar flag)

---

## **TEXT\_NOALARMES**

```
public static final java.lang.String TEXT_NOALARMES
```

Text d'informació d'historial buit

---

## **TEXT\_RESETHISTORIAL**

```
public static final java.lang.String TEXT_RESETHISTORIAL
```

Text de confirmació per a esborrar l'historial

---

## **TEXT\_DISPASSOCIAT**

```
public static final java.lang.String TEXT_DISPASSOCIAT
```



(continued from last page)

Text d'informació de dispositiu associat correctament

---

## TEXT\_SENSCANVIADA

```
public static final java.lang.String TEXT_SENSCANVIADA
```

Text d'informació de sensibilitat modificada correctament

---

## TEXT\_TERMINALROTAT

```
public static final java.lang.String TEXT_TERMINALROTAT
```

Text d'informació de dispositiu associat correctament

---

## TEXT\_TRANSFERIRHISTORIAL

```
public static final java.lang.String TEXT_TRANSFERIRHISTORIAL
```

Text d'informació de transferir historial

---

## TEXT\_NOBTACTIU

```
public static final java.lang.String TEXT_NOBTACTIU
```

Text d'informació de connectivitat bluetooth desactivada

---

## TEXT\_HISTTRANSFERIT

```
public static final java.lang.String TEXT_HISTTRANSFERIT
```

Text d'informació d'historial d'alarmes transferit correctament

---

## TEXT\_BUSQUEDAPROGRES

```
public static final java.lang.String TEXT_BUSQUEDAPROGRES
```

Text d'informació de búsqueda en progrès

---

# Package Dades

Conté les classes per a gestionar i emmagatzemar totes les dades de l'aplicació

# Dades

## Class BD

java.lang.Object

└─Dades.BD

public class **BD**  
extends java.lang.Object

Classe que actua de gestor de Base de Dades de l'aplicació

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

private	<a href="#">RMS</a> Objecte de Record Management Store
---------	---

### Constructor Summary

public	<a href="#">BD()</a> Crea una instància de BD
--------	--

### Method Summary

void	<a href="#">delConfig()</a> Esborra la configuració emmagatzemada a la BD
void	<a href="#">delHistorial()</a> Esborra l'historial d'alarmes emmagatzemat a la BD
<a href="#">Config</a>	<a href="#">getConfig()</a> Obté la configuració de l'aplicació emmagatzemada a la BD
<a href="#">Historial</a>	<a href="#">getHistorial()</a> Obté l'historial d'alarmes emmagatzemat a la BD
void	<a href="#">setConfig(<a href="#">Config</a> C)</a> Emmagatzema la configuració de l'aplicació a la BD
void	<a href="#">setHistorial(<a href="#">Historial</a> H)</a> Emmagatzema l'historial d'alarmes a la BD

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

## RMS

```
private Dades.RMS RMS
```

Objecte de Record Management Store

## Constructors

### BD

```
public BD( )
```

Crea una instància de BD

## Methods

### getConfig

```
public Config getConfig( )
```

Obté la configuració de l'aplicació emmagatzemada a la BD

**Returns:**

Configuració recuperada

### setConfig

```
public void setConfig(Config C)
```

Emmagatzema la configuració de l'aplicació a la BD

**Parameters:**

C - Configuració a emmagatzemar

### delConfig

```
public void delConfig( )
```

Esborra la configuració emmagatzemada a la BD

### getHistorial

```
public Historial getHistorial( )
```

Obté l'historial d'alarmes emmagatzemat a la BD

**Returns:**

Historial d'Alarmes recuperat

### setHistorial

```
public void setHistorial(Historial H)
```

Emmagatzema l'historial d'alarmes a la BD

**Parameters:**

H - Historial d'Alarmes a emmagatzemar

---

## **delHistorial**

```
public void delHistorial()
```

Esborra l'historial d'alarmes emmagatzemat a la BD

## Dades Class Config

```
java.lang.Object
├--Dades.Config
```

```
public class Config
extends java.lang.Object
```

Classe representativa de la configuració de l'aplicació

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

private	<a href="#"><u>bateria</u></a> Estat de la bateria del dispositiu PPStop (%)
private	<a href="#"><u>codi</u></a> Codi PPStop
private	<a href="#"><u>estat</u></a> Identificador de l'estat del servei PPStop
private	<a href="#"><u>idioma</u></a> Idioma de la interfície de l'aplicació
private	<a href="#"><u>maxtempsalarma</u></a> Temps màxim d'alarma (ms)
private	<a href="#"><u>ordre</u></a> Ordre del terminal en el sistema
private	<a href="#"><u>sensibilitat</u></a> Sensibilitat del sensor d'humitat del dispositiu PPStop (1-5)
private	<a href="#"><u>tipusalarma</u></a> Identificador del tpus d'alarma de l'aplicació
private	<a href="#"><u>volumalarma</u></a> Nivell de volum de l'alarma (%)

### Constructor Summary

public	<a href="#"><u>Config</u></a> ( ) Crea una instància de Config, amb la configuració per defecte
--------	--

### Method Summary

void	<a href="#"><u>fromByteStream</u></a> (byte[] dades) Des-Serialitza l'objecte
------	--

int	<a href="#"><u>getBateria()</u></a> Obté l'estat de la bateria del dispositiu PPStop (%)
java.lang.String	<a href="#"><u>getCodi()</u></a> Obté el codi PPStop
int	<a href="#"><u>getEstat()</u></a> Obté l'estat del servei PPStop
int	<a href="#"><u>getIdioma()</u></a> Obté l'idioma de la interfície de l'aplicació
int	<a href="#"><u>getMaxTempsAlarma()</u></a> Obté el temps màxim d'alarma
int	<a href="#"><u>getOrdre()</u></a> Obte l'ordre del terminal en el sistema
int	<a href="#"><u>getSensibilitat()</u></a> Obte la sensibilitat del sensor d'humitat del dispositiu PPStop
int	<a href="#"><u>getTipusAlarma()</u></a> Obté el tipus d'alarma de l'aplicació
int	<a href="#"><u>getVolumAlarma()</u></a> Obté el volum de l'alarma
void	<a href="#"><u>setBateria(int valor)</u></a> Guarda l'estat de la bateria del dispositiu PPStop (%)
void	<a href="#"><u>setCodi(java.lang.String valor)</u></a> Guarda el codi PPStop
void	<a href="#"><u>setEstat(int valor)</u></a> Guarda l'estat del servei PPStop
void	<a href="#"><u>setIdioma(int idioma)</u></a> Guardar l'idioma de la interfície de l'aplicació
void	<a href="#"><u>setMaxTempsAlarma(int valor)</u></a> Guarda el temps màxim d'alarma
void	<a href="#"><u>setOrdre(int valor)</u></a> Guarda l'ordre del terminal en el sistema
void	<a href="#"><u>setSensibilitat(int valor)</u></a> Guarda la sensibilitat del sensor d'humitat del dispositiu PPStop
void	<a href="#"><u>setTipusAlarma(int valor)</u></a> Guarda el tipus d'alarma de l'aplicació
void	<a href="#"><u>setVolumAlarma(int valor)</u></a> Guarda el volum de l'alarma
byte[]	<a href="#"><u>toByteArray()</u></a> Serialitza l'objecte

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

---

## Fields

### **codi**

```
private java.lang.String codi
```

Codi PPStop

---

### **estat**

```
private int estat
```

Identificador de l'estat del servei PPStop

---

### **bateria**

```
private int bateria
```

Estat de la bateria del dispositiu PPStop (%)

---

### **sensibilitat**

```
private int sensibilitat
```

Sensibilitat del sensor d'humitat del dispositiu PPStop (1-5)

---

### **ordre**

```
private int ordre
```

Ordre del terminal en el sistema

---

### **idioma**

```
private int idioma
```

Idioma de la interfície de l'aplicació

---

### **tipusalarma**

```
private int tipusalarma
```

Identificador del tpus d'alarma de l'aplicació

---

### **maxtempsalarma**

```
private int maxtempsalarma
```

Temps màxim d'alarma (ms)

---



(continued from last page)

## volumalarma

```
private int volumalarma
```

Nivell de volum de l'alarma (%)

## Constructors

### Config

```
public Config()
```

Crea una instància de Config, amb la configuració per defecte

## Methods

### getCodi

```
public java.lang.String getCodi()
```

Obté el codi PPStop

**Returns:**

Codi PPStop

### setCodi

```
public void setCodi(java.lang.String valor)
```

Guarda el codi PPStop

**Parameters:**

valor - Codi PPStop

### getEstat

```
public int getEstat()
```

Obté l'estat del servei PPStop

**Returns:**

Identificador de l'estat del servei

### setEstat

```
public void setEstat(int valor)
```

Guarda l'estat del servei PPStop

**Parameters:**

valor - Identificador de l'estat del servei

### getBateria

```
public int getBateria()
```

Obté l'estat de la bateria del dispositiu PPStop (%)

(continued from last page)

**Returns:**

Estat de la bateria (%)

---

**setBateria**

```
public void setBateria(int valor)
```

Guarda l'estat de la bateria del dispositiu PPSTop (%)

**Parameters:**

valor - Estat de la bateria (%)

---

**getSensibilitat**

```
public int getSensibilitat()
```

Obte la sensibilitat del sensor d'humitat del dispositiu PPSTop

**Returns:**

Sensibilitat del sensor (1-5)

---

**setSensibilitat**

```
public void setSensibilitat(int valor)
```

Guarda la sensibilitat del sensor d'humitat del dispositiu PPSTop

**Parameters:**

valor - Sensibilitat del sensor (0:baix/1:alt)

---

**getOrdre**

```
public int getOrdre()
```

Obte l'ordre del terminal en el sistema

**Returns:**

Ordre del terminal (cardinal)

---

**setOrdre**

```
public void setOrdre(int valor)
```

Guarda l'ordre del terminal en el sistema

**Parameters:**

valor - Ordre del terminal (cardinal)

---

**getIdioma**

```
public int getIdioma()
```

Obté l'idioma de la interfície de l'aplicació

**Returns:**

Identificador de l'idioma

(continued from last page)

---

## setIdioma

```
public void setIdioma(int idioma)
```

Guardar l'idioma de la interfície de l'aplicació

**Parameters:**

idioma - Identificador de l'idioma

---

## getTipusAlarma

```
public int getTipusAlarma()
```

Obté el tipus d'alarma de l'aplicació

**Returns:**

Identificador del tipus d'alarma

---

## setTipusAlarma

```
public void setTipusAlarma(int valor)
```

Guarda el tipus d'alarma de l'aplicació

**Parameters:**

valor - Identificador del tipus d'alarma

---

## getMaxTempsAlarma

```
public int getMaxTempsAlarma()
```

Obté el temps màxim d'alarma

**Returns:**

Temps màxim d'alarma (ms)

---

## setMaxTempsAlarma

```
public void setMaxTempsAlarma(int valor)
```

Guarda el temps màxim d'alarma

**Parameters:**

valor - Temps màxim d'alarma (ms)

---

## getVolumAlarma

```
public int getVolumAlarma()
```

Obté el volum de l'alarma

**Returns:**

Nivell de volum de l'alarma (%)

---

## setVolumAlarma

```
public void setVolumAlarma(int valor)
```

---

(continued from last page)

Guarda el volum de l'alarma

**Parameters:**

valor - Nivell de volum de l'alarma (%)

---

## toByteArray

```
public byte[] toByteArray()  
    throws java.io.IOException
```

Serialitza l'objecte

**Returns:**

Stream de bytes (Objecte serialitzat)

---

## fromByteArray

```
public void fromByteArray(byte[] dades)  
    throws java.io.IOException
```

Des-Serialitza l'objecte

**Parameters:**

dades - Stream de bytes (Objecte des-serialitzat)

# Dades

## Class Historial

```
java.lang.Object
|
+--Dades.Historial
```

```
public class Historial
extends java.lang.Object
```

Classe representativa de l'Historial d'Alarmes

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
3.1 (Maig de 2010)

### Field Summary

private	<a href="#">v</a> Vector de timestamps
---------	---

### Constructor Summary

public	<a href="#">Historial()</a> Crea una instància de Historial
--------	--

### Method Summary

void	<a href="#">delHistorial()</a> Reseteja l'Historial d'Alarmes
void	<a href="#">fromByteArray(byte[] dades)</a> Des-Serialitza l'objecte
java.util.Vector	<a href="#">getHistorial()</a> Obté el vector amb l'Historial d'Alarmes
void	<a href="#">setHistorial(java.util.Vector valor)</a> Guarda el vector amb l'Historial d'Alarmes
void	<a href="#">setNovaAlarma(java.lang.String valor)</a> Afegeix una nova alarma a l'Historial
byte[]	<a href="#">toByteArray()</a> Serialitza l'objecte

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

(continued from last page)

## V

```
private java.util.Vector v
```

Vector de timestamps

## Constructors

### Historial

```
public Historial()
```

Crea una instància de Historial

## Methods

### getHistorial

```
public java.util.Vector getHistorial()
```

Obté el vector amb l'Historial d'Alarmes

**Returns:**

Vector de timestamps

### setHistorial

```
public void setHistorial(java.util.Vector valor)
```

Guarda el vector amb l'Historial d'Alarmes

**Parameters:**

valor - Vector de timestamps

### setNovaAlarma

```
public void setNovaAlarma(java.lang.String valor)
```

Afegeix una nova alarma a l'Historial

**Parameters:**

valor - Timestamp de l'alarma a afegir

### delHistorial

```
public void delHistorial()
```

Reseteja l'Historial d'Alarmes

### toByteStream

```
public byte[] toByteStream()  
throws java.io.IOException
```

Serialitza l'objecte

**Returns:**

(continued from last page)

Stream de bytes (Objecte serialitzat)

---

## fromByteArray

```
public void fromByteArray(byte[] dades)  
    throws java.io.IOException
```

Des-Serialitza l'objecte

### Parameters:

dades - Stream de bytes (Objecte des-serialitzat)

## Dades Class Model

```
java.lang.Object
├--Dades.Model
```

```
public class Model
extends java.lang.Object
```

Classe que conté totes les dades de l'aplicació. Actua com a Model dins del patró MVC (Model-View-Controller)

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

private	<a href="#">A</a> Referència a l'Alarma Sonora
private	<a href="#">BD</a> Objecte Gestor de la BD
private	<a href="#">C</a> Objecte que conté la Configuració
private	<a href="#">estatBT</a> Identificador de l'estat de la connectivitat bluetooth del terminal
private	<a href="#">flag</a> Flag de servei
private	<a href="#">H</a> Objecte que conté l'Historial d'alarmes
private	<a href="#">macMobil</a> Adreça bluetooth del terminal
private	<a href="#">refresh</a> Indica si la Vista ha de ser refrescada
private	<a href="#">sensibilitatTX</a> Nova sensibilitat del sensor d'humitat del dispositiu PPStop (1-5) a TX
private	<a href="#">U</a> Objecte amb funcions d'Utilitat per al tractament de dades
private	<a href="#">V</a> Referència a la Vista

### Constructor Summary

public	<a href="#">Model</a> ( ) Crea una instància de Model, amb els valors per defecte
--------	--



## Method Summary

int	<a href="#"><u>getBateria()</u></a> Obté l'estat de la bateria del dispositiu PPStop (%)
java.lang.String	<a href="#"><u>getCodi()</u></a> Obté el codi PPStop
java.lang.String	<a href="#"><u>getDarreraAlarma()</u></a> Obté la darrera alarma de l'Historial d'Alarmes en format TXT
java.lang.String	<a href="#"><u>getDarreraAlarmaDif()</u></a> Obté la diferència de temps amb la darrera alarma de l'Historial d'Alarmes en format TXT
int	<a href="#"><u>getEstat()</u></a> Obte l'estat del servei PPStop
int	<a href="#"><u>getEstatBT()</u></a> Obté l'estat de la connectivitat Bluetooth del terminal
int	<a href="#"><u>getFlag()</u></a> Obté el flag de servei
<a href="#"><u>Historial</u></a>	<a href="#"><u>getHistorial()</u></a> Obté l'objecte Historial d'Alarmes
java.util.Vector	<a href="#"><u>getHistorialTXT()</u></a> Obté l'Historial d'Alarmes en format TXT
int	<a href="#"><u>getIdioma()</u></a> Obté l'idioma de la interfície de l'aplicació
java.lang.String	<a href="#"><u>getMacMovil()</u></a> Obté l'adreça bluetooth del terminal mòbil
int	<a href="#"><u>getMaxTempsAlarma()</u></a> Obté el temps màxim d'alarma
int	<a href="#"><u>getOrdre()</u></a> Obté l'ordre del terminal en el sistema
boolean	<a href="#"><u>getRefresh()</u></a> Obté si la vista ha de ser refrescada
int	<a href="#"><u>getSensibilitat()</u></a> Obté la sensibilitat del sensor d'humitat del dispositiu PPStop
int	<a href="#"><u>getSensibilitatTX()</u></a> Obté la nova sensibilitat del sensor d'humitat a TX
int	<a href="#"><u>getTipusAlarma()</u></a> Obté el tipus d'alarma de l'aplicació
int	<a href="#"><u>getVolumAlarma()</u></a> Obté el volum de l'alarma
void	<a href="#"><u>registrarAlarma()</u></a> Registra una nova alarma a l'Historial d'Alarmes

void	<a href="#"><code>resetHistorial()</code></a> Reseteja l'Historial d'Alarmes
void	<a href="#"><code>resetValors()</code></a> Reseteja els valors del model als valors per defecte
void	<a href="#"><code>setAlarma(Alarma A)</code></a> Associa l'alarma al model (patró MVC)
void	<a href="#"><code>setBateria(int valor)</code></a> Guarda l'estat de la bateria del dispositiu PPStop (%)
void	<a href="#"><code>setCodi(java.lang.String valor)</code></a> Estableix el codi PPStop
void	<a href="#"><code>setEstat(int valor)</code></a> Estableix l'estat del servei PPStop
void	<a href="#"><code>setEstatBT(int valor)</code></a> Estableix l'estat de la connectivitat Bluetooth del terminal
void	<a href="#"><code>setFlag(int valor)</code></a> Estableix el flag de servei
void	<a href="#"><code>setIdioma(int valor)</code></a> Estableix l'idioma de la interfície de l'aplicació
void	<a href="#"><code>setMacMovil(java.lang.String valor)</code></a> Guarda l'adreça bluetooth del terminal mòbil
void	<a href="#"><code>setMaxTempsAlarma(int valor)</code></a> Estableix el temps màxim d'alarma (ms)
void	<a href="#"><code>setOrdre(int valor)</code></a> Guarda l'ordre del terminal en el sistema
void	<a href="#"><code>setParametres(int bateria, int sensibilitat, int ordre)</code></a> Guarda els paràmetres generals del sistema: estat de la bateria, nivell de sensibilitat, ordre del terminal en el sistema
void	<a href="#"><code>setRefresh(boolean valor)</code></a> Estableix si la vista ha de ser refrescada
void	<a href="#"><code>setSensibilitat(int valor)</code></a> Guarda la sensibilitat del sensor d'humitat del dispositiu PPStop
void	<a href="#"><code>setSensibilitatTX(int valor)</code></a> Estableix la nova sensibilitat del sensor d'humitat a TX
void	<a href="#"><code>setTipusAlarma(int valor)</code></a> Estableix el tipus d'alarma de l'aplicació
void	<a href="#"><code>setVista(Vista V)</code></a> Associa la vista al model (patró MVC)
void	<a href="#"><code>setVolumAlarma(int valor)</code></a> Estableix el volum de l'alarma

**Methods inherited from class** `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,  
toString, wait, wait, wait
```

## Fields

### V

```
private Vista.Vista V
```

Referència a la Vista

### A

```
private Audio.Alarma A
```

Referència a l'Alarma Sonora

### C

```
private Dades.Config C
```

Objecte que conté la Configuració

### H

```
private Dades.Historial H
```

Objecte que conté l'Historial d'alarmes

### BD

```
private Dades.BD BD
```

Objecte Gestor de la BD

### U

```
private Utils.Utils U
```

Objecte amb funcions d'Utilitat per al tractament de dades

### macMobil

```
private java.lang.String macMobil
```

Adreça bluetooth del terminal

### flag

```
private int flag
```

Flag de servei

(continued from last page)

---

## estatBT

```
private int estatBT
```

Identificador de l'estat de la connectivitat bluetooth del terminal

---

## sensibilitatTX

```
private int sensibilitatTX
```

Nova sensibilitat del sensor d'humitat del dispositiu PPStop (1-5) a TX

---

## refresh

```
private boolean refresh
```

Indica si la Vista ha de ser refrescada

## Constructors

### Model

```
public Model()
```

Crea una instància de Model, amb els valors per defecte

## Methods

### setVista

```
public void setVista(Vista V)
```

Associa la vista al model (patró MVC)

**Parameters:**

v - Referència a la Vista

---

### setAlarma

```
public void setAlarma(Alarma A)
```

Associa l'alarma al model (patró MVC)

**Parameters:**

A - Referència a l'Alarma Sonora

---

### getMacMovel

```
public java.lang.String getMacMovel()
```

Obté l'adreça bluetooth del terminal mòbil

**Returns:**

Adreça bluetooth

---

(continued from last page)

---

## setMacMovel

```
public void setMacMovel(java.lang.String valor)
```

Guarda l'adreça bluetooth del terminal mòbil

### Parameters:

valor - Adreça bluetooth

---

## getFlag

```
public int getFlag()
```

Obté el flag de servei

### Returns:

Identificador del flag de servei

---

## setFlag

```
public void setFlag(int valor)
```

Estableix el flag de servei

### Parameters:

valor - Identificador del flag de servei

---

## getEstatBT

```
public int getEstatBT()
```

Obté l'estat de la connectivitat Bluetooth del terminal

### Returns:

Identificador de l'estat de la connectivitat

---

## setEstatBT

```
public void setEstatBT(int valor)
```

Estableix l'estat de la connectivitat Bluetooth del terminal

### Parameters:

valor - Identificador de l'estat de la connectivitat

---

## getEstat

```
public int getEstat()
```

Obte l'estat del servei PPStop

### Returns:

Identificador de l'estat del servei

---

## setEstat

```
public void setEstat(int valor)
```

---

(continued from last page)

Estableix l'estat del servei PPStop

**Parameters:**

valor - Identificador de l'estat del servei

---

## getBateria

```
public int getBateria()
```

Obté l'estat de la bateria del dispositiu PPStop (%)

**Returns:**

Estat de la bateria (%)

---

## setBateria

```
public void setBateria(int valor)
```

Guarda l'estat de la bateria del dispositiu PPStop (%)

**Parameters:**

valor - Estat de la bateria (%)

---

## getSensibilitat

```
public int getSensibilitat()
```

Obté la sensibilitat del sensor d'humitat del dispositiu PPStop

**Returns:**

Sensibilitat del sensor (0:baix/1:alt)

---

## setSensibilitat

```
public void setSensibilitat(int valor)
```

Guarda la sensibilitat del sensor d'humitat del dispositiu PPStop

**Parameters:**

valor - Sensibilitat del sensor (0:baix/1:alt)

---

## getSensibilitatTX

```
public int getSensibilitatTX()
```

Obté la nova sensibilitat del sensor d'humitat a TX

**Returns:**

Sensibilitat del sensor a TX (0:baix/1:alt)

---

## setSensibilitatTX

```
public void setSensibilitatTX(int valor)
```

Estableix la nova sensibilitat del sensor d'humitat a TX

**Parameters:**

valor - Sensibilitat del sensor a TX (0:baix/1:alt)

---

## getOrdre

```
public int getOrdre()
```

Obté l'ordre del terminal en el sistema

**Returns:**

Ordre del terminal (cardinal)

---

## setOrdre

```
public void setOrdre(int valor)
```

Guarda l'ordre del terminal en el sistema

**Parameters:**

valor - Ordre del terminal (cardinal)

---

## setParametres

```
public void setParametres(int bateria,  
    int sensibilitat,  
    int ordre)
```

Guarda els paràmetres generals del sistema: estat de la bateria, nivell de sensibilitat, ordre del terminal en el sistema

**Parameters:**

valor - Estat de la bateria (%)

valor - Sensibilitat del sensor (0:baix/1:alt)

valor - Ordre del terminal (cardinal)

---

## getCodi

```
public java.lang.String getCodi()
```

Obté el codi PPStop

**Returns:**

Codi PPStop

---

## setCodi

```
public void setCodi(java.lang.String valor)
```

Estableix el codi PPStop

**Parameters:**

valor - Codi PPStop

---

## getIdioma

```
public int getIdioma()
```

Obté l'idioma de la interfície de l'aplicació

**Returns:**

Identificador de l'idioma

---

---

## setIdioma

```
public void setIdioma(int valor)
```

Estableix l'idioma de la interfície de l'aplicació

**Parameters:**

valor - Identificador de l'idioma

---

## getTipusAlarma

```
public int getTipusAlarma()
```

Obté el tipus d'alarma de l'aplicació

**Returns:**

Identificador del tipus d'alarma

---

## setTipusAlarma

```
public void setTipusAlarma(int valor)
```

Estableix el tipus d'alarma de l'aplicació

**Parameters:**

valor - Identificador del tipus d'alarma

---

## getMaxTempsAlarma

```
public int getMaxTempsAlarma()
```

Obté el temps màxim d'alarma

**Returns:**

Temps màxim d'alarma (ms)

---

## setMaxTempsAlarma

```
public void setMaxTempsAlarma(int valor)
```

Estableix el temps màxim d'alarma (ms)

**Parameters:**

valor - Temps màxim d'alarma (ms)

---

## getVolumAlarma

```
public int getVolumAlarma()
```

Obté el volum de l'alarma

**Returns:**

Nivell de volum de l'alarma (%)

---

## setVolumAlarma

```
public void setVolumAlarma(int valor)
```

---



(continued from last page)

Estableix el volum de l'alarma

**Parameters:**

valor - Nivell de volum de l'alarma (%)

---

## getRefresh

```
public boolean getRefresh()
```

Obté si la vista ha de ser refrescada

**Returns:**

Valor booleà

---

## setRefresh

```
public void setRefresh(boolean valor)
```

Estableix si la vista ha de ser refrescada

**Parameters:**

valor - Valor booleà

---

## resetValors

```
private void resetValors()
```

Reseteja els valors del model als valors per defecte

---

## getHistorial

```
public Historial getHistorial()
```

Obté l'objecte Historial d'Alarmes

**Returns:**

Objecte Historial

---

## getHistorialTXT

```
public java.util.Vector getHistorialTXT()
```

Obté l'Historial d'Alarmes en format TXT

**Returns:**

Vector d'alarmes en format TXT

---

## getDarreraAlarma

```
public java.lang.String getDarreraAlarma()
```

Obté la darrera alarma de l'Historial d'Alarmes en format TXT

**Returns:**

Darrera alarma en format TXT

(continued from last page)

## **getDarreraAlarmaDif**

```
public java.lang.String getDarreraAlarmaDif()
```

Obté la diferència de temps amb la darrera alarma de l'Historial d'Alarmes en format TXT

### **Returns:**

Diferència de temps en format TXT

---

## **registrarAlarma**

```
public void registrarAlarma()
```

Registra una nova alarma a l'Historial d'Alarmes

---

## **resetHistorial**

```
public void resetHistorial()
```

Reseteja l'Historial d'Alarmes

# Dades

## Class RMS

```
java.lang.Object
```

```
└--Dades.RMS
```

```
public class RMS
extends java.lang.Object
```

Classe que gestiona la RMS (Record Manager Store)

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Field Summary

private	<a href="#">RS</a> Objecte de Record Store
---------	---

### Constructor Summary

public	<a href="#">RMS()</a> Crea una instància de RMS
--------	--

### Method Summary

void	<a href="#">closeRS()</a> Tanca un Record Store
void	<a href="#">delRS(java.lang.String RS)</a> Elimina un Record Store
byte[]	<a href="#">getRS(java.lang.String RS)</a> Llegeix un flux de bytes del Record Store
void	<a href="#">openRS(java.lang.String RS)</a> Obre un Record Store
byte[]	<a href="#">readRecord()</a> Llegeix un Record del Record Store obert
void	<a href="#">setRS(java.lang.String RS, byte[] b)</a> Guarda un flux de bytes al Record Store
void	<a href="#">writeRecord(byte[] b)</a> Escriu un Record en el Record Store obert

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### RS

```
private javax.microedition.rms.RecordStore RS
```

Objecte de Record Store

## Constructors

### RMS

```
public RMS()
```

Crea una instància de RMS

## Methods

### openRS

```
private void openRS(java.lang.String RS)
```

Obre un Record Store

**Parameters:**

RS - Identificador del Record Store a obrir

---

### closeRS

```
private void closeRS()
```

Tanca un Record Store

---

### writeRecord

```
private void writeRecord(byte[] b)
```

Escriu un Record en el Record Store obert

**Parameters:**

b - Flux de dades a escriure

---

### readRecord

```
private byte[] readRecord()
```

Llegeix un Record del Record Store obert

**Returns:**

Flux de dades llegit

---

### getRS

```
public byte[] getRS(java.lang.String RS)
```

Llegeix un flux de bytes del Record Store

(continued from last page)

**Parameters:**

RS - Record Store sobre el que aplicar la lectura

**Returns:**Flux de bytes llegits

---

**setRS**

```
public void setRS(java.lang.String RS,  
                 byte[] b)
```

Guarda un flux de bytes al Record Store

**Parameters:**

RS - Record Store sobre el que aplicar la escriptura

b - Flux de bytes a escriure

---

**delRS**

```
public void delRS(java.lang.String RS)
```

Elimina un Record Store

**Parameters:**RS - Record Store a eliminar

---

---

## Package Main

Conté la classe principal de l'aplicació

# Main Class PPStop

```

java.lang.Object
  |
  +-- javax.microedition.midlet.MIDlet
        |
        +-- Main.PPStop
  
```

public class **PPStop**  
 extends javax.microedition.midlet.MIDlet

Classe principal de l'aplicació PPStop

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

## Nested Class Summary

class	<a href="#">PPStop.endTimerAlarma</a> PPStop.endTimerAlarma
-------	--

## Field Summary

private	<a href="#">A</a> Alarma Sonora
private	<a href="#">M</a> Model de dades
private	<a href="#">S</a> Servei PPStop
private	<a href="#">T</a> Objecte per a timeouts
private	<a href="#">V</a> Vista del MIDlet

## Fields inherited from class javax.microedition.midlet.MIDlet

state
-------

## Constructor Summary

public	<a href="#">PPStop()</a> Crea una instància de PPStop
--------	--

## Method Summary

void	<a href="#">destroyApp(boolean unconditional)</a> Sortir del MIDlet: destruir-lo
------	---

void	<a href="#"><code>exitMIDlet()</code></a> Notifica la sortida del MIDlet
void	<a href="#"><code>initMIDlet()</code></a> Inicialitza el servei bluetooth PPStop
void	<a href="#"><code>log(java.lang.String msg)</code></a> Mètode auxiliar de LOG per a funcions de Debugging
void	<a href="#"><code>mostrarDialogInformatiu(int tipus, java.lang.String text, long timeout, int height)</code></a> Mostra un dialog informatiu durant un cert timeout
void	<a href="#"><code>pauseApp()</code></a> Notifica la pausa del MIDlet
void	<a href="#"><code>PlayAudio(java.lang.String arxiu, java.lang.String tipus)</code></a> Reprodueix instantàniament un arxiu d'audio
void	<a href="#"><code>StartAlarma()</code></a> Activa l'alarma PPStop
void	<a href="#"><code>startApp()</code></a> Crea tots els elements del patró Model-View-Controller (MVC) i inicialitza el MIDlet
void	<a href="#"><code>StopAlarma()</code></a> Atura l'alarma PPStop

#### Methods inherited from class `javax.microedition.midlet.MIDlet`

`checkPermission`, `destroyApp`, `getAppProperty`, `getProxy`, `notifyDestroyed`, `notifyPaused`, `pauseApp`, `platformRequest`, `resumeRequest`, `startApp`

#### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`, `toString`, `wait`, `wait`, `wait`

## Fields

### M

`private Dades.Model M`

Model de dades

### V

`private Vista.Vista V`

Vista del MIDlet

### A

`private Audio.Alarma A`

Alarma Sonora



## S

```
private Bluetooth.Servei S
```

Servei PPStop

---

## T

```
private java.util.Timer T
```

Objecte per a timeouts

---

## Constructors

### PPStop

```
public PPStop()
```

Crea una instància de PPStop

---

## Methods

### startApp

```
public void startApp()
```

Crea tots els elements del patró Model-View-Controller (MVC) i inicialitza el MIDlet

---

### StartAlarma

```
public void StartAlarma()
```

Activa l'alarma PPStop

---

### StopAlarma

```
public void StopAlarma()
```

Atura l'alarma PPStop

---

### PlayAudio

```
public void PlayAudio(java.lang.String arxiu,  
    java.lang.String tipus)
```

Reprodueix instantàniament un arxiu d'audio

#### Parameters:

arxiu - Arxiu d'audio a reproduir

tipus - Tipus d'arxiu d'àudio a reproduir

---

### mostrarDialogInformatiu

```
public void mostrarDialogInformatiu(int tipus,  
    java.lang.String text,  
    long timeout,  
    int height)
```

---

(continued from last page)

Mostra un dialog informatiu durant un cert timeout

**Parameters:**

tipus - Tipus de Dialog a mostrar  
text - Contingut del Dialog  
timeout - Timeout del Dialog (ms)  
height - Alçada del Dialog (px)

---

**initMIDlet**

```
private void initMIDlet()
```

Inicialitza el servei bluetooth PPStop

---

**exitMIDlet**

```
public void exitMIDlet()
```

Notifica la sortida del MIDlet

---

**pauseApp**

```
public void pauseApp()
```

Notifica la pausa del MIDlet

---

**destroyApp**

```
public void destroyApp(boolean unconditional)
```

Sortir del MIDlet: destruir-lo

---

**log**

```
public void log(java.lang.String msg)
```

Mètode auxiliar de LOG per a funcions de Debugging

**Parameters:**

msg - Missatge a mostrar per pantalla

---

# Main

## Class PPStop.endTimerAlarma

```

java.lang.Object
  |
  +- java.util.TimerTask
      |
      +- Main.PPStop.endTimerAlarma

```

### All Implemented Interfaces:

```
java.lang.Runnable
```

```

private class PPStop.endTimerAlarma
extends java.util.TimerTask

```

Final del Timer d'Alarma PPStop: Atura l'alarma

### Fields inherited from class java.util.TimerTask

CANCELLED, EXECUTED, lock, nextExecutionTime, period, SCHEDULED, state, VIRGIN

## Constructor Summary

private	<a href="#">PPStop.endTimerAlarma()</a>
---------	---

## Method Summary

void	<a href="#">run()</a>
------	-----------------------

### Methods inherited from class java.util.TimerTask

cancel, run, scheduledExecutionTime

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Methods inherited from interface java.lang.Runnable

run

## Constructors

### PPStop.endTimerAlarma

```
private PPStop.endTimerAlarma()
```

## Methods

(continued from last page)

## **run**

```
public void run()
```

---

# Package Utils

Conté les classes amb funcions auxiliars de certa utilitat

## Utils

### Class Utils

```
java.lang.Object
|
+--Utils.Utils
```

```
public class Utils
extends java.lang.Object
```

Classe amb funcions útils d'àmbit general

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
3.1 (Maig de 2010)

### Constructor Summary

public	<a href="#"><code>Utils()</code></a> Crea una instància de Utils
--------	---

### Method Summary

java.util.Vector	<a href="#"><code>invertirVector</code></a> (java.util.Vector vec) Inverteix els elements d'un vector
java.lang.String	<a href="#"><code>obtenirData</code></a> (long timestamp, int format) Obté la data en un cert format a partir d'un timestamp
java.lang.String	<a href="#"><code>obtenirDiferenciaData</code></a> (long timestamp, int idioma) Obté la diferència entre la data actual i una data passada en format TXT
long	<a href="#"><code>obtenirTimestamp</code></a> () Obté el timestamps actual
static java.lang.String	<a href="#"><code>substituirStr</code></a> (java.lang.String text, java.lang.String comodi, java.lang.String substitut) Reemplaça strings d'una cadena de caràcters

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Constructors

#### Utils

```
public Utils()
```

Crea una instància de Utils

### Methods

---

(continued from last page)

## obtenirTimestamp

```
public long obtenirTimestamp()
```

Obte el timestamps actual

### Returns:

Timestamp actual

---

## obtenirData

```
public java.lang.String obtenirData(long timestamp,
    int format)
```

Obté la data en un cert format a partir d'un timestamp

### Parameters:

timestamp - Timestamp a convertir

format - Format de la data a obtenir

### Returns:

Data formatada

---

## obtenirDiferenciaData

```
public java.lang.String obtenirDiferenciaData(long timestamp,
    int idioma)
```

Obté la diferència entre la data actual i una data passada en format TXT

### Parameters:

timestamp - Timestamp de la data passada

idioma - Idioma desitjat

### Returns:

Diferència en format TXT

---

## invertirVector

```
public java.util.Vector invertirVector(java.util.Vector vec)
```

Inverteix els elements d'un vector

### Parameters:

vec - Vector a invertir

### Returns:

Vector invertit

---

## substituirStr

```
public static java.lang.String substituirStr(java.lang.String text,
    java.lang.String comodi,
    java.lang.String substitut)
```

Reemplaça strings d'una cadena de caràcters

### Parameters:

(continued from last page)

`text` - Text sobre el qual s'aplica la substitució

`comodi` - Cadena de text a substituir

`substitut` - Cadena de text substituïda

**Returns:**

Text amb la substitució aplicada



## Utils

### Class UtilsVista

```
java.lang.Object
|
+--Utils.UtilsVista
```

```
public class UtilsVista
extends java.lang.Object
```

Classe amb funcions útils en l'àmbit de la creació de la GUI (Graphic User Interface)

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

3.1 (Maig de 2010)

### Constructor Summary

public	<a href="#"><u>UtilsVista()</u></a> Crea una instància de UtilsVista
--------	---

### Method Summary

com.sun.lwuit.ComboBox	<a href="#"><u>crearComboBox</u></a> (java.lang.String[] opcions, int labelW) Crea un ComboBox amb unes determinades opcions
com.sun.lwuit.Container	<a href="#"><u>crearParell</u></a> (java.lang.String label, java.lang.String text, int labelW) Crea un contenedor amb un parell label<->text
com.sun.lwuit.TextArea	<a href="#"><u>crearTextArea</u></a> (java.lang.String text, int labelW, boolean editable, boolean focusable) Crea un TextArea amb unes determinades característiques
com.sun.lwuit.TextArea	<a href="#"><u>crearTextField</u></a> (java.lang.String text, int labelW, boolean editable, boolean focusable, int tipus, int maxsize) Crea un TextField amb unes determinades característiques

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Constructors

### UtilsVista

```
public UtilsVista()
```

Crea una instància de UtilsVista

## Methods

(continued from last page)

---

## crearComboBox

```
public com.sun.lwuit.ComboBox crearComboBox(java.lang.String[] opciones,  
int labelW)
```

Crea un ComboBox amb unes determinades opcions

### Parameters:

opcions - Vector d'opcions a mostrar en el combobox  
labelW - Amplada del ComboBox (px)

### Returns:

ComboBox creat

---

## crearTextArea

```
public com.sun.lwuit.TextArea crearTextArea(java.lang.String text,  
int labelW,  
boolean editable,  
boolean focusable)
```

Crea un TextArea amb unes determinades característiques

### Parameters:

text - Text que contindrà el TextArea  
labelW - Amplada del TextArea (px)  
editable - Indica si el contingut del TextArea ha de ser editable  
focusable - Indica si el TextArea ha de poder rebre un event de focus

### Returns:

TextArea creat

---

## crearTextField

```
public com.sun.lwuit.TextArea crearTextField(java.lang.String text,  
int labelW,  
boolean editable,  
boolean focusable,  
int tipus,  
int maxsize)
```

Crea un TextField amb unes determinades característiques

### Parameters:

text - Text que contindrà el TextField  
labelW - Amplada del TextField (px)  
editable - Indica si el contingut del TextField ha de ser editable  
focusable - Indica si el TextField ha de poder rebre un event de focus  
tipus - Indica el tipus de restricció del contingut del TextField  
maxsize - Indica el tamany màxim assumible pel TextField

### Returns:

TextField creat

---

## crearParell

```
public com.sun.lwuit.Container crearParell(java.lang.String label,  
java.lang.String text,  
int labelW)
```

(continued from last page)

Crea un contenedor amb un parell `label<->text`

**Parameters:**

`label` - Etiqueta de text del parell

`text` - Text que contindrà el TextArea del parell

`labelw` - Amplada de cada un dels elements del parell (px)

**Returns:**

Contenedor creat

---

# Package Vista

Conté les classes per a generar i gestionar la GUI (Graphic User Interface) de l'aplicació

# Vista

## Class FormAlarma

```

java.lang.Object
  |
  +--com.sun.lwuit.Component
        |
        +--com.sun.lwuit.Container
              |
              +--com.sun.lwuit.Form
                    |
                    +--Vista.FormAlarma
  
```

### All Implemented Interfaces:

com.sun.lwuit.events.StyleListener, com.sun.lwuit.animations.Animation

```

public class FormAlarma
extends com.sun.lwuit.Form
  
```

Classe que defineix els elements gràfics del formulari d'Alarma

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

3.1 (Maig de 2010)

#### Fields inherited from class com.sun.lwuit.Form

animatableComponents, backCommand, backSK, backspaceSK, clearCommand, clearSK, commandListener, contentPane, current, cyclicFocus, defaultCommand, dragged, focusCycleRoot, focusDownSequence, focused, focusRightSequence, gameKeyListeners, glassPane, keyListeners, lastFocused, leftSK, mediaComponents, menuBar, previousForm, rightSK, rightSK2, selectCommand, tint, tintColor, title, transitionInAnimator, transitionOutAnimator

#### Fields inherited from class com.sun.lwuit.Container

cmpTransitions, components, layout, scrollableX, scrollableY, shouldLayout

#### Fields inherited from class com.sun.lwuit.Component

animationMotion, animationSpeed, beforeLastScrollX, beforeLastScrollY, BOTTOM, bounds, BRB\_CENTER\_OFFSET, BRB\_CONSTANT\_ASCENT, BRB\_CONSTANT\_DESCENT, BRB\_OTHER, CACHE\_MISS\_THREASHOLD, cellRenderer, CENTER, clientProperties, componentLabel, destScrollY, dirtyRegion, dirtyRegionLock, dragActivated, draggedMotion, enabled, fixedPosition, focusable, focused, focusListeners, focusPainted, handlesInput, id, initialized, initialScrollY, isScrollVisible, lastDragged, lastScrollX, lastScrollY, lastTime, LEFT, nextFocusDown, nextFocusLeft, nextFocusRight, nextFocusUp, painterBounds, parent, pLastDragged, preferredSize, RIGHT, scrollSize, scrollSizeRequestedByUser, scrollX, scrollY, selectedStyle, selectText, shouldCalcPreferredSize, shouldCalcScrollSize, sizeRequestedByUser, smoothScrolling, TOP, unSelectedStyle, visible

## Constructor Summary

public	<a href="#">FormAlarma</a> ( <a href="#">Vista</a> V, <a href="#">Model</a> M, int idioma)
	Crea una instància de FormAlarma

#### Methods inherited from class com.sun.lwuit.Form

actionCommand, actionCommandImpl, addCommand, addCommand, addCommandListener, addComponent, addComponent, addComponent, addComponent, addGameKeyListener, addKeyListener, addKeyListener, addSelectCommand, addSortedComponent, addSortedComponentDown, addSortedComponentRight, clearFocusVectors, createCommandList, createMenuCancelCommand, createMenuSelectCommand, createSelectCommand, deregisterAnimated, deregisterMediaComponent, dispose, disposeImpl, findAllFocusable, findScrollableAncestor, fireCommand, fireKeyEvent, focusChangeTriggersRevalidate, getBackCommand, getClearCommand, getCommand, getCommandCount, getComponentForm, getComponentIndex, getContentPane, getDefaultCommand, getFocused, getFocusPosition, getGlassPane, getLayout, getMenuStyle, getPreviousForm, getScrollAnimationSpeed, getSoftButton, getSoftButtonCount, getSoftButtonStyle, getTintColor, getTitle, getTitleComponent, getTitleStyle, getTransitionInAnimator, getTransitionOutAnimator, hasAnimations, hasMedia, hideMenu, hideNotify, hidePopups, initComponentsImpl, initFocusDown, initFocused, initFocusRight, initLaf, isCyclicFocus, isInSameColumn, isInSameRow, isMenu, isReverseSoftButtons, isSingleFocusMode, isSmoothScrolling, keyPressed, keyReleased, keyRepeated, longKeyPress, longPointerPress, onShow, onShowCompleted, paint, paintBackground, paramString, pointerDragged, pointerHover, pointerPressed, pointerReleased, refreshTheme, registerAnimated, registerMediaComponent, removeAll, removeAllCommands, removeCommand, removeCommandListener, removeComponent, removeGameKeyListener, removeKeyListener, removeKeyListener, removeSelectCommand, repaint, repaintAnimations, replace, replaceAndWait, requestFocus, restoreMenu, scrollComponentToVisible, setBackCommand, setBgImage, setClearCommand, setCommandListener, setCyclicFocus, setDefaultCommand, setDraggedComponent, setFocused, setGlassPane, setLayout, setMenuCellRenderer, setMenuStyle, setMenuTransitions, setScrollable, setScrollableX, setScrollableY, setScrollAnimationSpeed, setSmoothScrolling, setSoftButtonStyle, setTintColor, setTitle, setTitleComponent, setTitleComponent, setTitleStyle, setTransitionInAnimator, setTransitionOutAnimator, setVisible, show, show, showBack, showDialog, showMenuDialog, showModal, showModal, showNotify, showPopup, sizeChanged, sizeChangedInternal, updateFocus

#### Methods inherited from class `com.sun.lwuit.Container`

addComponent, addComponent, addComponent, addComponent, calcPreferredSize, contains, deinitializeImpl, doLayout, flushReplace, getComponentAt, getComponentAt, getComponentCount, getComponentIndex, getComponentsNames, getLayout, getLayoutHeight, getLayoutWidth, getScrollableParent, initComponentsImpl, insertComponentAt, invalidate, isParentOf, isScrollableX, isScrollableY, layoutContainer, paint, paintIntersecting, paramString, pointerHover, pointerPressed, refreshTheme, removeAll, removeComponent, replace, replace, replaceAndWait, replaceComponents, requestFocusChild, revalidate, scrollComponentToVisible, setCellRenderer, setLayout, setScrollable, setScrollableX, setScrollableY, setShouldCalcPreferredSize

#### Methods inherited from class `com.sun.lwuit.Component`

```

addFocusListener, animate, calcPreferredSize, calcScrollSize, checkAnimation,
contains, deinitialize, deinitializeCustomStyle, deinitializeImpl, drawPainters,
fireActionEvent, fireClicked, fireFocusGained, fireFocusGained, fireFocusLost,
fireFocusLost, focusGained, focusGainedInternal, focusLost, focusLostInternal,
getAbsoluteX, getAbsoluteY, getAnimationMotion, getBaseline,
getBaselineResizeBehavior, getBorder, getBottomGap, getBounds, getClientProperty,
getComponentForm, getDirtyRegion, getHeight, getLabelForComponent, getNextFocusDown,
getNextFocusLeft, getNextFocusRight, getNextFocusUp, getParent, getPreferredH,
getPreferredSize, getPreferredW, getScrollAnimationSpeed, getScrollDimension,
getScrollX, getScrollY, getSelectCommandText, getSelectedStyle, getSideGap, getStyle,
getUIID, getUnselectedStyle, getWidth, getX, getY, handlesInput, hasFocus,
initComponent, initComponentImpl, initCustomStyle, initScrollMotion, initStyle,
isBorderPainted, isCellRenderer, isDragActivated, isEnabled, isFixedPosition,
isFocusable, isFocusPainted, isInitialized, isScrollable, isScrollableX,
isScrollableY, isScrollVisible, isSelectableInteraction, isSmoothScrolling,
isVisible, keyPressed, keyReleased, keyRepeated, laidOut, longKeyPress,
longPointerPress, mergeStyle, onEditComplete, paint, paintBackground,
paintBackgrounds, paintBorder, paintComponent, paintComponent, paintInternal,
paintInternal, paintIntersectingComponentsAbove, paintScrollbars, paintScrollbarX,
paintScrollbarY, paramString, pointerDragged, pointerDragged, pointerHover,
pointerHoverReleased, pointerPressed, pointerPressed, pointerReleased,
pointerReleased, preferredSize, putClientProperty, refreshTheme, refreshTheme,
removeFocusListener, repaint, repaint, repaint, requestFocus, scrollRectToVisible,
setAnimationMotion, setBorderPainted, setCellRenderer, setDirtyRegion,
setDragActivated, setEnabled, setFixedPosition, setFocus, setFocusable,
setFocusPainted, setHandlesInput, setHeight, setInitialized, setIsScrollVisible,
setLabelForComponent, setNextFocusDown, setNextFocusLeft, setNextFocusRight,
setNextFocusUp, setParent, setPreferredH, setPreferredSize, setPreferredW,
setScrollAnimationSpeed, setScrollSize, setScrollX, setScrollY, setSelectCommandText,
setSelectedStyle, setShouldCalcPreferredSize, setSize, setSmoothScrolling, setStyle,
setUIID, setUnselectedStyle, setVisible, setWidth, setX, setY, styleChanged, toString

```

#### Methods inherited from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait

```

#### Methods inherited from interface `com.sun.lwuit.animations.Animation`

```

animate, paint

```

#### Methods inherited from interface `com.sun.lwuit.events.StyleListener`

```

styleChanged

```

## Constructors

### FormAlarma

```

public FormAlarma(Vista V,
                  Model M,
                  int idioma)

```

Crea una instància de FormAlarma

(continued from last page)

**Parameters:**

V - Referència a la Vista

M - Referència al Model de dades

idioma - Identificador de l'idioma



# Vista

## Class FormConfig

```

java.lang.Object
  |
  +--com.sun.lwuit.Component
        |
        +--com.sun.lwuit.Container
              |
              +--com.sun.lwuit.Form
                    |
                    +--Vista.FormConfig
  
```

### All Implemented Interfaces:

com.sun.lwuit.events.StyleListener, com.sun.lwuit.animations.Animation

```

public class FormConfig
extends com.sun.lwuit.Form
  
```

Classe que defineix els elements gràfics del formulari de Configuració

### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

### Version:

3.1 (Maig de 2010)

## Field Summary

private	<a href="#">UV</a> Objecte per a Utilitats de generació de la Vista
---------	--

### Fields inherited from class com.sun.lwuit.Form

animatableComponents, backCommand, backSK, backspaceSK, clearCommand, clearSK, commandListener, contentPane, current, cyclicFocus, defaultCommand, dragged, focusCycleRoot, focusDownSequence, focused, focusRightSequence, gameKeyListeners, glassPane, keyListeners, lastFocused, leftSK, mediaComponents, menuBar, previousForm, rightSK, rightSK2, selectCommand, tint, tintColor, title, transitionInAnimator, transitionOutAnimator

### Fields inherited from class com.sun.lwuit.Container

cmpTransitions, components, layout, scrollableX, scrollableY, shouldLayout

### Fields inherited from class com.sun.lwuit.Component

animationMotion, animationSpeed, beforeLastScrollX, beforeLastScrollY, BOTTOM, bounds, BRB\_CENTER\_OFFSET, BRB\_CONSTANT\_ASCENT, BRB\_CONSTANT\_DESCENT, BRB\_OTHER, CACHE\_MISS\_THREASHOLD, cellRenderer, CENTER, clientProperties, componentLabel, destScrollY, dirtyRegion, dirtyRegionLock, dragActivated, draggedMotion, enabled, fixedPosition, focusable, focused, focusListeners, focusPainted, handlesInput, id, initialized, initialScrollY, isScrollVisible, lastDragged, lastScrollX, lastScrollY, lastTime, LEFT, nextFocusDown, nextFocusLeft, nextFocusRight, nextFocusUp, painterBounds, parent, pLastDragged, preferredSize, RIGHT, scrollSize, scrollSizeRequestedByUser, scrollX, scrollY, selectedStyle, selectText, shouldCalcPreferredSize, shouldCalcScrollSize, sizeRequestedByUser, smoothScrolling, TOP, unSelectedStyle, visible

## Constructor Summary

public

[FormConfig](#)([Vista](#) V, [Model](#) M, int idioma)

Crea una instància de FormConfig

**Methods inherited from class** `com.sun.lwuit.Form`

actionCommand, actionCommandImpl, addCommand, addCommand, addCommandListener, addComponent, addComponent, addComponent, addComponent, addGameKeyListener, addKeyListener, addKeyListener, addSelectCommand, addSortedComponent, addSortedComponentDown, addSortedComponentRight, clearFocusVectors, createCommandList, createMenuCancelCommand, createMenuSelectCommand, createSelectCommand, deregisterAnimated, deregisterMediaComponent, dispose, disposeImpl, findAllFocusable, findScrollableAncestor, fireCommand, fireKeyEvent, focusChangeTriggersRevalidate, getBackCommand, getClearCommand, getCommand, getCommandCount, getComponentForm, getComponentIndex, getContentPane, getDefaultCommand, getFocused, getFocusPosition, getGlassPane, getLayout, getMenuStyle, getPreviousForm, getScrollAnimationSpeed, getSoftButton, getSoftButtonCount, getSoftButtonStyle, getTintColor, getTitle, getTitleComponent, getTitleStyle, getTransitionInAnimator, getTransitionOutAnimator, hasAnimations, hasMedia, hideMenu, hideNotify, hidePopups, initComponentsImpl, initFocusDown, initFocused, initFocusRight, initLaf, isCyclicFocus, isInSameColumn, isInSameRow, isMenu, isReverseSoftButtons, isSingleFocusMode, isSmoothScrolling, keyPressed, keyReleased, keyRepeated, longKeyPress, longPointerPress, onShow, onShowCompleted, paint, paintBackground, paramString, pointerDragged, pointerHover, pointerPressed, pointerReleased, refreshTheme, registerAnimated, registerMediaComponent, removeAll, removeAllCommands, removeCommand, removeCommandListener, removeComponent, removeGameKeyListener, removeKeyListener, removeKeyListener, removeSelectCommand, repaint, repaintAnimations, replace, replaceAndWait, requestFocus, restoreMenu, scrollComponentToVisible, setBackCommand, setBgImage, setClearCommand, setCommandListener, setCyclicFocus, setDefaultCommand, setDraggedComponent, setFocused, setGlassPane, setLayout, setMenuCellRenderer, setMenuStyle, setMenuTransitions, setScrollable, setScrollableX, setScrollableY, setScrollAnimationSpeed, setSmoothScrolling, setSoftButtonStyle, setTintColor, setTitle, setTitleComponent, setTitleComponent, setTitleStyle, setTransitionInAnimator, setTransitionOutAnimator, setVisible, show, show, showBack, showDialog, showMenuDialog, showModal, showModal, showNotify, showPopup, sizeChanged, sizeChangedInternal, updateFocus

**Methods inherited from class** `com.sun.lwuit.Container`

addComponent, addComponent, addComponent, addComponent, calcPreferredSize, contains, deinitializeImpl, doLayout, flushReplace, getComponentAt, getComponentAt, getComponentCount, getComponentIndex, getComponentsNames, getLayout, getLayoutHeight, getLayoutWidth, getScrollableParent, initComponentsImpl, insertComponentAt, invalidate, isParentOf, isScrollableX, isScrollableY, layoutContainer, paint, paintIntersecting, paramString, pointerHover, pointerPressed, refreshTheme, removeAll, removeComponent, replace, replace, replaceAndWait, replaceComponents, requestFocusChild, revalidate, scrollComponentToVisible, setCellRenderer, setLayout, setScrollable, setScrollableX, setScrollableY, setShouldCalcPreferredSize

**Methods inherited from class** `com.sun.lwuit.Component`

```

addFocusListener, animate, calcPreferredSize, calcScrollSize, checkAnimation,
contains, deinitialize, deinitializeCustomStyle, deinitializeImpl, drawPainters,
fireActionEvent, fireClicked, fireFocusGained, fireFocusLost, fireFocusLost,
focusGained, focusGainedInternal, focusLost, focusLostInternal,
getAbsoluteX, getAbsoluteY, getAnimationMotion, getBaseline,
getBaselineResizeBehavior, getBorder, getBottomGap, getBounds, getClientProperty,
getComponentForm, getDirtyRegion, getHeight, getLabelForComponent, getNextFocusDown,
getNextFocusLeft, getNextFocusRight, getNextFocusUp, getParent, getPreferredH,
getPreferredSize, getPreferredW, getScrollAnimationSpeed, getScrollDimension,
getScrollX, getScrollY, getSelectCommandText, getSelectedStyle, getSideGap, getStyle,
getUIID, getUnselectedStyle, getWidth, getX, getY, handlesInput, hasFocus,
initComponent, initComponentImpl, initCustomStyle, initScrollMotion, initStyle,
isBorderPainted, isCellRenderer, isDragActivated, isEnabled, isFixedPosition,
isFocusable, isFocusPainted, isInitialized, isScrollable, isScrollableX,
isScrollableY, isScrollVisible, isSelectableInteraction, isSmoothScrolling,
isVisible, keyPressed, keyReleased, keyRepeated, laidOut, longKeyPress,
longPointerPress, mergeStyle, onEditComplete, paint, paintBackground,
paintBackgrounds, paintBorder, paintComponent, paintComponent, paintInternal,
paintInternal, paintIntersectingComponentsAbove, paintScrollbars, paintScrollbarX,
paintScrollbarY, paramString, pointerDragged, pointerDragged, pointerHover,
pointerHoverReleased, pointerPressed, pointerPressed, pointerReleased,
pointerReleased, preferredSize, putClientProperty, refreshTheme, refreshTheme,
removeFocusListener, repaint, repaint, repaint, requestFocus, scrollRectToVisible,
setAnimationMotion, setBorderPainted, setCellRenderer, setDirtyRegion,
setDragActivated, setEnabled, setFixedPosition, setFocus, setFocusable,
setFocusPainted, setHandlesInput, setHeight, setInitialized, setIsScrollVisible,
setLabelForComponent, setNextFocusDown, setNextFocusLeft, setNextFocusRight,
setNextFocusUp, setParent, setPreferredH, setPreferredSize, setPreferredW,
setScrollAnimationSpeed, setScrollSize, setScrollX, setScrollY, setSelectCommandText,
setSelectedStyle, setShouldCalcPreferredSize, setSize, setSmoothScrolling, setStyle,
setUIID, setUnselectedStyle, setVisible, setWidth, setX, setY, styleChanged, toString

```

#### Methods inherited from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait

```

#### Methods inherited from interface `com.sun.lwuit.animations.Animation`

```

animate, paint

```

#### Methods inherited from interface `com.sun.lwuit.events.StyleListener`

```

styleChanged

```

## Fields

### UV

```
private Utils.UtilsVista UV
```

Objecte per a Utilitats de generació de la Vista

## Constructors

(continued from last page)

## FormConfig

```
public FormConfig(Vista V,  
                  Model M,  
                  int idioma)
```

Crea una instància de FormConfig

### Parameters:

V - Referència a la Vista

M - Referència al Model de dades

idioma - Identificador de l'idioma

# Vista

## Class FormHistorial

```

java.lang.Object
  |
  +--com.sun.lwuit.Component
        |
        +--com.sun.lwuit.Container
              |
              +--com.sun.lwuit.Form
                    |
                    +--Vista.FormHistorial
  
```

### All Implemented Interfaces:

com.sun.lwuit.events.StyleListener, com.sun.lwuit.animations.Animation

```

public class FormHistorial
extends com.sun.lwuit.Form
  
```

Classe que defineix els elements gràfics del formulari d'Historial d'Alarmes

### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

### Version:

3.1 (Maig de 2010)

## Nested Class Summary

class	<a href="#">FormHistorial.LlistaRenderer</a> FormHistorial.LlistaRenderer
-------	--

### Fields inherited from class com.sun.lwuit.Form

animatableComponents, backCommand, backSK, backspaceSK, clearCommand, clearSK, commandListener, contentPane, current, cyclicFocus, defaultCommand, dragged, focusCycleRoot, focusDownSequence, focused, focusRightSequence, gameKeyListeners, glassPane, keyListeners, lastFocused, leftSK, mediaComponents, menuBar, previousForm, rightSK, rightSK2, selectCommand, tint, tintColor, title, transitionInAnimator, transitionOutAnimator

### Fields inherited from class com.sun.lwuit.Container

cmpTransitions, components, layout, scrollableX, scrollableY, shouldLayout

### Fields inherited from class com.sun.lwuit.Component

animationMotion, animationSpeed, beforeLastScrollX, beforeLastScrollY, BOTTOM, bounds, BRB\_CENTER\_OFFSET, BRB\_CONSTANT\_ASCENT, BRB\_CONSTANT\_DESCENT, BRB\_OTHER, CACHE\_MISS\_THREASHOLD, cellRenderer, CENTER, clientProperties, componentLabel, destScrollY, dirtyRegion, dirtyRegionLock, dragActivated, draggedMotion, enabled, fixedPosition, focusable, focused, focusListeners, focusPainted, handlesInput, id, initialized, initialScrollY, isScrollVisible, lastDragged, lastScrollX, lastScrollY, lastTime, LEFT, nextFocusDown, nextFocusLeft, nextFocusRight, nextFocusUp, painterBounds, parent, pLastDragged, preferredSize, RIGHT, scrollSize, scrollSizeRequestedByUser, scrollX, scrollY, selectedStyle, selectText, shouldCalcPreferredSize, shouldCalcScrollSize, sizeRequestedByUser, smoothScrolling, TOP, unSelectedStyle, visible

## Constructor Summary

public

[FormHistorial](#)([Vista](#) V, [Model](#) M, int idioma)

Crea una instància de FormHistorial

**Methods inherited from class** `com.sun.lwuit.Form`

actionCommand, actionCommandImpl, addCommand, addCommand, addCommandListener, addComponent, addComponent, addComponent, addComponent, addGameKeyListener, addKeyListener, addKeyListener, addSelectCommand, addSortedComponent, addSortedComponentDown, addSortedComponentRight, clearFocusVectors, createCommandList, createMenuCancelCommand, createMenuSelectCommand, createSelectCommand, deregisterAnimated, deregisterMediaComponent, dispose, disposeImpl, findAllFocusable, findScrollableAncestor, fireCommand, fireKeyEvent, focusChangeTriggersRevalidate, getBackCommand, getClearCommand, getCommand, getCommandCount, getComponentForm, getComponentIndex, getContentPane, getDefaultCommand, getFocused, getFocusPosition, getGlassPane, getLayout, getMenuStyle, getPreviousForm, getScrollAnimationSpeed, getSoftButton, getSoftButtonCount, getSoftButtonStyle, getTintColor, getTitle, getTitleComponent, getTitleStyle, getTransitionInAnimator, getTransitionOutAnimator, hasAnimations, hasMedia, hideMenu, hideNotify, hidePopups, initComponentsImpl, initFocusDown, initFocused, initFocusRight, initLaf, isCyclicFocus, isInSameColumn, isInSameRow, isMenu, isReverseSoftButtons, isSingleFocusMode, isSmoothScrolling, keyPressed, keyReleased, keyRepeated, longKeyPress, longPointerPress, onShow, onShowCompleted, paint, paintBackground, paramString, pointerDragged, pointerHover, pointerPressed, pointerReleased, refreshTheme, registerAnimated, registerMediaComponent, removeAll, removeAllCommands, removeCommand, removeCommandListener, removeComponent, removeGameKeyListener, removeKeyListener, removeKeyListener, removeSelectCommand, repaint, repaintAnimations, replace, replaceAndWait, requestFocus, restoreMenu, scrollComponentToVisible, setBackCommand, setBgImage, setClearCommand, setCommandListener, setCyclicFocus, setDefaultCommand, setDraggedComponent, setFocused, setGlassPane, setLayout, setMenuCellRenderer, setMenuStyle, setMenuTransitions, setScrollable, setScrollableX, setScrollableY, setScrollAnimationSpeed, setSmoothScrolling, setSoftButtonStyle, setTintColor, setTitle, setTitleComponent, setTitleComponent, setTitleStyle, setTransitionInAnimator, setTransitionOutAnimator, setVisible, show, show, showBack, showDialog, showMenuDialog, showModal, showModal, showNotify, showPopup, sizeChanged, sizeChangedInternal, updateFocus

**Methods inherited from class** `com.sun.lwuit.Container`

addComponent, addComponent, addComponent, addComponent, calcPreferredSize, contains, deinitializeImpl, doLayout, flushReplace, getComponentAt, getComponentAt, getComponentCount, getComponentIndex, getComponentsNames, getLayout, getLayoutHeight, getLayoutWidth, getScrollableParent, initComponentsImpl, insertComponentAt, invalidate, isParentOf, isScrollableX, isScrollableY, layoutContainer, paint, paintIntersecting, paramString, pointerHover, pointerPressed, refreshTheme, removeAll, removeComponent, replace, replace, replaceAndWait, replaceComponents, requestFocusChild, revalidate, scrollComponentToVisible, setCellRenderer, setLayout, setScrollable, setScrollableX, setScrollableY, setShouldCalcPreferredSize

**Methods inherited from class** `com.sun.lwuit.Component`

```

addFocusListener, animate, calcPreferredSize, calcScrollSize, checkAnimation,
contains, deinitialize, deinitializeCustomStyle, deinitializeImpl, drawPainters,
fireActionEvent, fireClicked, fireFocusGained, fireFocusGained, fireFocusLost,
fireFocusLost, focusGained, focusGainedInternal, focusLost, focusLostInternal,
getAbsoluteX, getAbsoluteY, getAnimationMotion, getBaseline,
getBaselineResizeBehavior, getBorder, getBottomGap, getBounds, getClientProperty,
getComponentForm, getDirtyRegion, getHeight, getLabelForComponent, getNextFocusDown,
getNextFocusLeft, getNextFocusRight, getNextFocusUp, getParent, getPreferredH,
getPreferredSize, getPreferredW, getScrollAnimationSpeed, getScrollDimension,
getScrollX, getScrollY, getSelectCommandText, getSelectedStyle, getSideGap, getStyle,
getUIID, getUnselectedStyle, getWidth, getX, getY, handlesInput, hasFocus,
initComponent, initComponentImpl, initCustomStyle, initScrollMotion, initStyle,
isBorderPainted, isCellRenderer, isDragActivated, isEnabled, isFixedPosition,
isFocusable, isFocusPainted, isInitialized, isScrollable, isScrollableX,
isScrollableY, isScrollVisible, isSelectableInteraction, isSmoothScrolling,
isVisible, keyPressed, keyReleased, keyRepeated, laidOut, longKeyPress,
longPointerPress, mergeStyle, onEditComplete, paint, paintBackground,
paintBackgrounds, paintBorder, paintComponent, paintComponent, paintInternal,
paintInternal, paintIntersectingComponentsAbove, paintScrollbars, paintScrollbarX,
paintScrollbarY, paramString, pointerDragged, pointerDragged, pointerHover,
pointerHoverReleased, pointerPressed, pointerPressed, pointerReleased,
pointerReleased, preferredSize, putClientProperty, refreshTheme, refreshTheme,
removeFocusListener, repaint, repaint, repaint, requestFocus, scrollRectToVisible,
setAnimationMotion, setBorderPainted, setCellRenderer, setDirtyRegion,
setDragActivated, setEnabled, setFixedPosition, setFocus, setFocusable,
setFocusPainted, setHandlesInput, setHeight, setInitialized, setIsScrollVisible,
setLabelForComponent, setNextFocusDown, setNextFocusLeft, setNextFocusRight,
setNextFocusUp, setParent, setPreferredH, setPreferredSize, setPreferredW,
setScrollAnimationSpeed, setScrollSize, setScrollX, setScrollY, setSelectCommandText,
setSelectedStyle, setShouldCalcPreferredSize, setSize, setSmoothScrolling, setStyle,
setUIID, setUnselectedStyle, setVisible, setWidth, setX, setY, styleChanged, toString

```

#### Methods inherited from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait

```

#### Methods inherited from interface `com.sun.lwuit.animations.Animation`

```

animate, paint

```

#### Methods inherited from interface `com.sun.lwuit.events.StyleListener`

```

styleChanged

```

## Constructors

### FormHistorial

```

public FormHistorial(Vista V,
                    Model M,
                    int idioma)

```

Crea una instància de FormHistorial

(continued from last page)

**Parameters:**

v - Referència a la Vista

m - Referència al Model de dades

idioma - Identificador de l'idioma



# Vista

## Class FormHistorial.LlistaRenderer

```

java.lang.Object
  |
  +--com.sun.lwuit.Component
        |
        +--com.sun.lwuit.Container
              |
              +--Vista.FormHistorial.LlistaRenderer

```

### All Implemented Interfaces:

com.sun.lwuit.list.ListCellRenderer, com.sun.lwuit.events.StyleListener, com.sun.lwuit.animations.Animation

### class FormHistorial.LlistaRenderer

extends com.sun.lwuit.Container

implements com.sun.lwuit.animations.Animation, com.sun.lwuit.events.StyleListener, com.sun.lwuit.list.ListCellRenderer

Subclasse que defineix l'aspecte gràfic de la Llista

## Field Summary

private	<a href="#">alarma</a>	Etiqueta de text de cada element de la llista
private	<a href="#">focus</a>	Etiqueta de focus de cada element de la llista
private	<a href="#">icon</a>	Icona de cada element de la llista
private	<a href="#">imatge</a>	Etiqueta de background de cada element de la llista

### Fields inherited from class com.sun.lwuit.Container

cmpTransitions, components, layout, scrollableX, scrollableY, shouldLayout

### Fields inherited from class com.sun.lwuit.Component

animationMotion, animationSpeed, beforeLastScrollX, beforeLastScrollY, BOTTOM, bounds, BRB\_CENTER\_OFFSET, BRB\_CONSTANT\_ASCENT, BRB\_CONSTANT\_DESCENT, BRB\_OTHER, CACHE\_MISS\_THRESHOLD, cellRenderer, CENTER, clientProperties, componentLabel, destScrollY, dirtyRegion, dirtyRegionLock, dragActivated, draggedMotion, enabled, fixedPosition, focusable, focused, focusListeners, focusPainted, handlesInput, id, initialized, initialScrollY, isScrollVisible, lastDragged, lastScrollX, lastScrollY, lastTime, LEFT, nextFocusDown, nextFocusLeft, nextFocusRight, nextFocusUp, painterBounds, parent, pLastDragged, preferredSize, RIGHT, scrollSize, scrollSizeRequestedByUser, scrollX, scrollY, selectedStyle, selectText, shouldCalcPreferredSize, shouldCalcScrollSize, sizeRequestedByUser, smoothScrolling, TOP, unSelectedStyle, visible

## Constructor Summary

public	<a href="#">FormHistorial.LlistaRenderer()</a> Components de cada element de la Llista
--------	---

## Method Summary

com.sun.lwuit.Component	<a href="#">getListCellRendererComponent</a> (com.sun.lwuit.List list, java.lang.Object value, int index, boolean isSelected) Definició del valor de cada component de cada element de la llista
com.sun.lwuit.Component	<a href="#">getListFocusComponent</a> (com.sun.lwuit.List list) Definició de l'estil de focus d'un element de la llista

### Methods inherited from class com.sun.lwuit.Container

addComponent, addComponent, addComponent, addComponent, calcPreferredSize, contains, deinitializeImpl, doLayout, flushReplace, getComponentAt, getComponentAt, getComponentCount, getComponentIndex, getComponentsNames, getLayout, getLayoutHeight, getLayoutWidth, getScrollableParent, initComponentsImpl, insertComponentAt, invalidate, isParentOf, isScrollableX, isScrollableY, layoutContainer, paint, paintIntersecting, paramString, pointerHover, pointerPressed, refreshTheme, removeAll, removeComponent, replace, replace, replaceAndWait, replaceComponents, requestFocusChild, revalidate, scrollComponentToVisible, setCellRenderer, setLayout, setScrollable, setScrollableX, setScrollableY, setShouldCalcPreferredSize

### Methods inherited from class com.sun.lwuit.Component

addFocusListener, animate, calcPreferredSize, calcScrollSize, checkAnimation, contains, deinitialize, deinitializeCustomStyle, deinitializeImpl, drawPainters, fireActionEvent, fireClicked, fireFocusGained, fireFocusGained, fireFocusLost, fireFocusLost, focusGained, focusGainedInternal, focusLost, focusLostInternal, getAbsoluteX, getAbsoluteY, getAnimationMotion, getBaseline, getBaselineResizeBehavior, getBorder, getBottomGap, getBounds, getClientProperty, getComponentForm, getDirtyRegion, getHeight, getLabelForComponent, getNextFocusDown, getNextFocusLeft, getNextFocusRight, getNextFocusUp, getParent, getPreferredH, getPreferredSize, getPreferredW, getScrollAnimationSpeed, getScrollDimension, getScrollX, getScrollY, getSelectCommandText, getSelectedStyle, getSideGap, getStyle, getUIID, getUnselectedStyle, getWidth, getX, getY, handlesInput, hasFocus, initComponents, initComponentsImpl, initCustomStyle, initScrollMotion, initStyle, isBorderPainted, isCellRenderer, isDragActivated, isEnabled, isFixedPosition, isFocusable, isFocusPainted, isInitialized, isScrollable, isScrollableX, isScrollableY, isScrollVisible, isSelectableInteraction, isSmoothScrolling, isVisible, keyPressed, keyReleased, keyRepeated, laidOut, longKeyPress, longPointerPress, mergeStyle, onEditComplete, paint, paintBackground, paintBackgrounds, paintBorder, paintComponent, paintComponent, paintInternal, paintInternal, paintIntersectingComponentsAbove, paintScrollbars, paintScrollbarX, paintScrollbarY, paramString, pointerDragged, pointerDragged, pointerHover, pointerHoverReleased, pointerPressed, pointerPressed, pointerReleased, pointerReleased, preferredSize, putClientProperty, refreshTheme, refreshTheme, removeFocusListener, repaint, repaint, repaint, requestFocus, scrollRectToVisible, setAnimationMotion, setBorderPainted, setCellRenderer, setDirtyRegion, setDragActivated, setEnabled, setFixedPosition, setFocus, setFocusable, setFocusPainted, setHandlesInput, setHeight, setInitialized, setIsScrollVisible, setLabelForComponent, setNextFocusDown, setNextFocusLeft, setNextFocusRight, setNextFocusUp, setParent, setPreferredH, setPreferredSize, setPreferredW, setScrollAnimationSpeed, setScrollSize, setScrollX, setScrollY, setSelectCommandText, setSelectedStyle, setShouldCalcPreferredSize, setSize, setSmoothScrolling, setStyle, setUIID, setUnselectedStyle, setVisible, setWidth, setX, setY, styleChanged, toString

**Methods inherited from class** `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait`

**Methods inherited from interface** `com.sun.lwuit.animations.Animation`

`animate, paint`

**Methods inherited from interface** `com.sun.lwuit.events.StyleListener`

`styleChanged`

**Methods inherited from interface** `com.sun.lwuit.list.ListCellRenderer`

`getListCellRendererComponent, getListFocusComponent`

## Fields

### **imatge**

`private com.sun.lwuit.Label imatge`

Etiqueta de background de cada element de la llista

### **alarma**

`private com.sun.lwuit.Label alarma`

Etiqueta de text de cada element de la llista

### **focus**

`private com.sun.lwuit.Label focus`

Etiqueta de focus de cada element de la llista

### **icon**

`private com.sun.lwuit.Image icon`

Icona de cada element de la llista

## Constructors

### **FormHistorial.LlistaRenderer**

`public FormHistorial.LlistaRenderer()`

Components de cada element de la Llista

## Methods

(continued from last page)

## getListCellRendererComponent

```
public com.sun.lwuit.Component getListCellRendererComponent(com.sun.lwuit.List list,  
    java.lang.Object value,  
    int index,  
    boolean isSelected)
```

Definició del valor de cada component de cada element de la llista

### Parameters:

list - Llista a aplicar Render  
value - Valor de l'element de la llista del Render actual  
index - Index de l'element de la llista del Render actual  
isSelected - Indica si l'element està seleccionat o no

### Returns:

Component sobre el qual s'ha fet Render

---

## getListFocusComponent

```
public com.sun.lwuit.Component getListFocusComponent(com.sun.lwuit.List list)
```

Definició de l'estil de focus d'un element de la llista

### Parameters:

list - Llista a aplicar Render

### Returns:

Component enfocat sobre el qual s'ha fet render

# Vista

## Class FormMenu

```

java.lang.Object
  |
  +--com.sun.lwuit.Component
        |
        +--com.sun.lwuit.Container
              |
              +--com.sun.lwuit.Form
                    |
                    +--Vista.FormMenu
  
```

### All Implemented Interfaces:

com.sun.lwuit.events.StyleListener, com.sun.lwuit.animations.Animation

```

public class FormMenu
extends com.sun.lwuit.Form
  
```

Classe que defineix els elements gràfics del formulari de Menú Principal

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

3.1 (Maig de 2010)

## Field Summary

private	<a href="#"><u>lm</u></a> Etiqueta de text del menú
private	<a href="#"><u>v</u></a> Referència a la Vista

### Fields inherited from class com.sun.lwuit.Form

animatableComponents, backCommand, backSK, backspaceSK, clearCommand, clearSK, commandListener, contentPane, current, cyclicFocus, defaultCommand, dragged, focusCycleRoot, focusDownSequence, focused, focusRightSequence, gameKeyListeners, glassPane, keyListeners, lastFocused, leftSK, mediaComponents, menuBar, previousForm, rightSK, rightSK2, selectCommand, tint, tintColor, title, transitionInAnimator, transitionOutAnimator

### Fields inherited from class com.sun.lwuit.Container

cmpTransitions, components, layout, scrollableX, scrollableY, shouldLayout

### Fields inherited from class com.sun.lwuit.Component

animationMotion, animationSpeed, beforeLastScrollX, beforeLastScrollY, BOTTOM, bounds, BRB\_CENTER\_OFFSET, BRB\_CONSTANT\_ASCENT, BRB\_CONSTANT\_DESCENT, BRB\_OTHER, CACHE\_MISS\_THRESHOLD, cellRenderer, CENTER, clientProperties, componentLabel, destScrollY, dirtyRegion, dirtyRegionLock, dragActivated, draggedMotion, enabled, fixedPosition, focusable, focused, focusListeners, focusPainted, handlesInput, id, initialized, initialScrollY, isScrollVisible, lastDragged, lastScrollX, lastScrollY, lastTime, LEFT, nextFocusDown, nextFocusLeft, nextFocusRight, nextFocusUp, painterBounds, parent, pLastDragged, preferredSize, RIGHT, scrollSize, scrollSizeRequestedByUser, scrollX, scrollY, selectedStyle, selectText, shouldCalcPreferredSize, shouldCalcScrollSize, sizeRequestedByUser, smoothScrolling, TOP, unSelectedStyle, visible

## Constructor Summary

public	<a href="#">FormMenu</a> ( <a href="#">Vista</a> V, int idioma) Crea una instància de Form Menu
--------	--

## Method Summary

com.sun.lwuit.Button	<a href="#">crearBotoMenu</a> (int n, java.lang.String text, int formDesti) Crea un botó per al Men Principal, amb un text i destí determinats
----------------------	---

### Methods inherited from class com.sun.lwuit.Form

actionCommand, actionCommandImpl, addCommand, addCommand, addCommandListener, addComponent, addComponent, addComponent, addComponent, addGameKeyListener, addKeyListener, addKeyListener, addSelectCommand, addSortedComponent, addSortedComponentDown, addSortedComponentRight, clearFocusVectors, createCommandList, createMenuCancelCommand, createMenuSelectCommand, createSelectCommand, deregisterAnimated, deregisterMediaComponent, dispose, disposeImpl, findAllFocusable, findScrollableAncestor, fireCommand, fireKeyEvent, focusChangeTriggersRevalidate, getBackCommand, getClearCommand, getCommand, getCommandCount, getComponentForm, getComponentIndex, getContentPane, getDefaultCommand, getFocused, getFocusPosition, getGlassPane, getLayout, getMenuStyle, getPreviousForm, getScrollAnimationSpeed, getSoftButton, getSoftButtonCount, getSoftButtonStyle, getTintColor, getTitle, getTitleComponent, getTitleStyle, getTransitionInAnimator, getTransitionOutAnimator, hasAnimations, hasMedia, hideMenu, hideNotify, hidePopups, initComponentsImpl, initFocusDown, initFocused, initFocusRight, initLaf, isCyclicFocus, isInSameColumn, isInSameRow, isMenu, isReverseSoftButtons, isSingleFocusMode, isSmoothScrolling, keyPressed, keyReleased, keyRepeated, longKeyPress, longPointerPress, onShow, onShowCompleted, paint, paintBackground, paramString, pointerDragged, pointerHover, pointerPressed, pointerReleased, refreshTheme, registerAnimated, registerMediaComponent, removeAll, removeAllCommands, removeCommand, removeCommandListener, removeComponent, removeGameKeyListener, removeKeyListener, removeKeyListener, removeSelectCommand, repaint, repaintAnimations, replace, replaceAndWait, requestFocus, restoreMenu, scrollComponentToVisible, setBackCommand, setBgImage, setClearCommand, setCommandListener, setCyclicFocus, setDefaultCommand, setDraggedComponent, setFocused, setGlassPane, setLayout, setMenuCellRenderer, setMenuStyle, setMenuTransitions, setScrollable, setScrollableX, setScrollableY, setScrollAnimationSpeed, setSmoothScrolling, setSoftButtonStyle, setTintColor, setTitle, setTitleComponent, setTitleComponent, setTitleStyle, setTransitionInAnimator, setTransitionOutAnimator, setVisible, show, show, showBack, showDialog, showMenuDialog, showModal, showModal, showNotify, showPopup, sizeChanged, sizeChangedInternal, updateFocus

### Methods inherited from class com.sun.lwuit.Container

addComponent, addComponent, addComponent, addComponent, calcPreferredSize, contains, deinitializeImpl, doLayout, flushReplace, getComponentAt, getComponentAt, getComponentCount, getComponentIndex, getComponentsNames, getLayout, getLayoutHeight, getLayoutWidth, getScrollableParent, initComponentsImpl, insertComponentAt, invalidate, isParentOf, isScrollableX, isScrollableY, layoutContainer, paint, paintIntersecting, paramString, pointerHover, pointerPressed, refreshTheme, removeAll, removeComponent, replace, replace, replaceAndWait, replaceComponents, requestFocusChild, revalidate, scrollComponentToVisible, setCellRenderer, setLayout, setScrollable, setScrollableX, setScrollableY, setShouldCalcPreferredSize

**Methods inherited from class** `com.sun.lwuit.Component`

`addFocusListener`, `animate`, `calcPreferredSize`, `calcScrollSize`, `checkAnimation`,  
`contains`, `deinitialize`, `deinitializeCustomStyle`, `deinitializeImpl`, `drawPainters`,  
`fireActionEvent`, `fireClicked`, `fireFocusGained`, `fireFocusGained`, `fireFocusLost`,  
`fireFocusLost`, `focusGained`, `focusGainedInternal`, `focusLost`, `focusLostInternal`,  
`getAbsoluteX`, `getAbsoluteY`, `getAnimationMotion`, `getBaseline`,  
`getBaselineResizeBehavior`, `getBorder`, `getBottomGap`, `getBounds`, `getClientProperty`,  
`getComponentForm`, `getDirtyRegion`, `getHeight`, `getLabelForComponent`, `getNextFocusDown`,  
`getNextFocusLeft`, `getNextFocusRight`, `getNextFocusUp`, `getParent`, `getPreferredH`,  
`getPreferredSize`, `getPreferredW`, `getScrollAnimationSpeed`, `getScrollDimension`,  
`getScrollX`, `getScrollY`, `getSelectCommandText`, `getSelectedStyle`, `getSideGap`, `getStyle`,  
`getUIID`, `getUnselectedStyle`, `getWidth`, `getX`, `getY`, `handlesInput`, `hasFocus`,  
`initComponent`, `initComponentImpl`, `initCustomStyle`, `initScrollMotion`, `initStyle`,  
`isBorderPainted`, `isCellRenderer`, `isDragActivated`, `isEnabled`, `isFixedPosition`,  
`isFocusable`, `isFocusPainted`, `isInitialized`, `isScrollable`, `isScrollableX`,  
`isScrollableY`, `isScrollVisible`, `isSelectableInteraction`, `isSmoothScrolling`,  
`isVisible`, `keyPressed`, `keyReleased`, `keyRepeated`, `laidOut`, `longKeyPress`,  
`longPointerPress`, `mergeStyle`, `onEditComplete`, `paint`, `paintBackground`,  
`paintBackgrounds`, `paintBorder`, `paintComponent`, `paintComponent`, `paintInternal`,  
`paintInternal`, `paintIntersectingComponentsAbove`, `paintScrollbars`, `paintScrollbarX`,  
`paintScrollbarY`, `paramString`, `pointerDragged`, `pointerDragged`, `pointerHover`,  
`pointerHoverReleased`, `pointerPressed`, `pointerPressed`, `pointerReleased`,  
`pointerReleased`, `preferredSize`, `putClientProperty`, `refreshTheme`, `refreshTheme`,  
`removeFocusListener`, `repaint`, `repaint`, `repaint`, `requestFocus`, `scrollRectToVisible`,  
`setAnimationMotion`, `setBorderPainted`, `setCellRenderer`, `setDirtyRegion`,  
`setDragActivated`, `setEnabled`, `setFixedPosition`, `setFocus`, `setFocusable`,  
`setFocusPainted`, `setHandlesInput`, `setHeight`, `setInitialized`, `setIsScrollVisible`,  
`setLabelForComponent`, `setNextFocusDown`, `setNextFocusLeft`, `setNextFocusRight`,  
`setNextFocusUp`, `setParent`, `setPreferredH`, `setPreferredSize`, `setPreferredW`,  
`setScrollAnimationSpeed`, `setScrollSize`, `setScrollX`, `setScrollY`, `setSelectCommandText`,  
`setSelectedStyle`, `setShouldCalcPreferredSize`, `setSize`, `setSmoothScrolling`, `setStyle`,  
`setUIID`, `setUnselectedStyle`, `setVisible`, `setWidth`, `setX`, `setY`, `styleChanged`, `toString`

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`,  
`toString`, `wait`, `wait`, `wait`

**Methods inherited from interface** `com.sun.lwuit.animations.Animation`

`animate`, `paint`

**Methods inherited from interface** `com.sun.lwuit.events.StyleListener`

`styleChanged`

## Fields

### V

`private Vista.Vista` **V**

Referència a la Vista

(continued from last page)

## lm

```
private com.sun.lwuit.Label lm
```

Etiqueta de text del menú

## Constructors

### FormMenu

```
public FormMenu(Vista v,  
                int idioma)
```

Crea una instància de Form Menu

#### Parameters:

v - Referència a la Vista

idioma - Identificador de l'idioma

## Methods

### crearBotoMenu

```
private com.sun.lwuit.Button crearBotoMenu(int n,  
                                             java.lang.String text,  
                                             int formDesti)  
throws java.lang.Exception
```

Crea un botó per al Men Principal, amb un text i destí determinats

#### Parameters:

n - Identificador numèric del botó a crear

text - Text a aplicar a l'etiqueta del menú en l'event FocusGained

formDesti - Identificador del formulari destí del botó

#### Returns:

Botó creat



# Vista

## Class FormResum

```

java.lang.Object
  |
  +--com.sun.lwuit.Component
        |
        +--com.sun.lwuit.Container
              |
              +--com.sun.lwuit.Form
                    |
                    +--Vista.FormResum
  
```

### All Implemented Interfaces:

com.sun.lwuit.events.StyleListener, com.sun.lwuit.animations.Animation

```

public class FormResum
extends com.sun.lwuit.Form
  
```

Classe que defineix els elements gràfics del formulari de Resum

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

3.1 (Maig de 2010)

## Field Summary

private	<a href="#">UV</a> Objecte per a Utilitats de generació de la Vista
---------	--

### Fields inherited from class com.sun.lwuit.Form

animatableComponents, backCommand, backSK, backspaceSK, clearCommand, clearSK, commandListener, contentPane, current, cyclicFocus, defaultCommand, dragged, focusCycleRoot, focusDownSequence, focused, focusRightSequence, gameKeyListeners, glassPane, keyListeners, lastFocused, leftSK, mediaComponents, menuBar, previousForm, rightSK, rightSK2, selectCommand, tint, tintColor, title, transitionInAnimator, transitionOutAnimator

### Fields inherited from class com.sun.lwuit.Container

cmpTransitions, components, layout, scrollableX, scrollableY, shouldLayout

### Fields inherited from class com.sun.lwuit.Component

animationMotion, animationSpeed, beforeLastScrollX, beforeLastScrollY, BOTTOM, bounds, BRB\_CENTER\_OFFSET, BRB\_CONSTANT\_ASCENT, BRB\_CONSTANT\_DESCENT, BRB\_OTHER, CACHE\_MISS\_THREASHOLD, cellRenderer, CENTER, clientProperties, componentLabel, destScrollY, dirtyRegion, dirtyRegionLock, dragActivated, draggedMotion, enabled, fixedPosition, focusable, focused, focusListeners, focusPainted, handlesInput, id, initialized, initialScrollY, isScrollVisible, lastDragged, lastScrollX, lastScrollY, lastTime, LEFT, nextFocusDown, nextFocusLeft, nextFocusRight, nextFocusUp, painterBounds, parent, pLastDragged, preferredSize, RIGHT, scrollSize, scrollSizeRequestedByUser, scrollX, scrollY, selectedStyle, selectText, shouldCalcPreferredSize, shouldCalcScrollSize, sizeRequestedByUser, smoothScrolling, TOP, unSelectedStyle, visible

## Constructor Summary

public

[FormResum](#)([Vista](#) V, [Model](#) M, int idioma)

Crea una instància de FormResum

**Methods inherited from class** `com.sun.lwuit.Form`

actionCommand, actionCommandImpl, addCommand, addCommand, addCommandListener, addComponent, addComponent, addComponent, addComponent, addGameKeyListener, addKeyListener, addKeyListener, addSelectCommand, addSortedComponent, addSortedComponentDown, addSortedComponentRight, clearFocusVectors, createCommandList, createMenuCancelCommand, createMenuSelectCommand, createSelectCommand, deregisterAnimated, deregisterMediaComponent, dispose, disposeImpl, findAllFocusable, findScrollableAncestor, fireCommand, fireKeyEvent, focusChangeTriggersRevalidate, getBackCommand, getClearCommand, getCommand, getCommandCount, getComponentForm, getComponentIndex, getContentPane, getDefaultCommand, getFocused, getFocusPosition, getGlassPane, getLayout, getMenuStyle, getPreviousForm, getScrollAnimationSpeed, getSoftButton, getSoftButtonCount, getSoftButtonStyle, getTintColor, getTitle, getTitleComponent, getTitleStyle, getTransitionInAnimator, getTransitionOutAnimator, hasAnimations, hasMedia, hideMenu, hideNotify, hidePopups, initComponentsImpl, initFocusDown, initFocused, initFocusRight, initLaf, isCyclicFocus, isInSameColumn, isInSameRow, isMenu, isReverseSoftButtons, isSingleFocusMode, isSmoothScrolling, keyPressed, keyReleased, keyRepeated, longKeyPress, longPointerPress, onShow, onShowCompleted, paint, paintBackground, paramString, pointerDragged, pointerHover, pointerPressed, pointerReleased, refreshTheme, registerAnimated, registerMediaComponent, removeAll, removeAllCommands, removeCommand, removeCommandListener, removeComponent, removeGameKeyListener, removeKeyListener, removeKeyListener, removeSelectCommand, repaint, repaintAnimations, replace, replaceAndWait, requestFocus, restoreMenu, scrollComponentToVisible, setBackCommand, setBgImage, setClearCommand, setCommandListener, setCyclicFocus, setDefaultCommand, setDraggedComponent, setFocused, setGlassPane, setLayout, setMenuCellRenderer, setMenuStyle, setMenuTransitions, setScrollable, setScrollableX, setScrollableY, setScrollAnimationSpeed, setSmoothScrolling, setSoftButtonStyle, setTintColor, setTitle, setTitleComponent, setTitleComponent, setTitleStyle, setTransitionInAnimator, setTransitionOutAnimator, setVisible, show, show, showBack, showDialog, showMenuDialog, showModal, showModal, showNotify, showPopup, sizeChanged, sizeChangedInternal, updateFocus

**Methods inherited from class** `com.sun.lwuit.Container`

addComponent, addComponent, addComponent, addComponent, calcPreferredSize, contains, deinitializeImpl, doLayout, flushReplace, getComponentAt, getComponentAt, getComponentCount, getComponentIndex, getComponentsNames, getLayout, getLayoutHeight, getLayoutWidth, getScrollableParent, initComponentsImpl, insertComponentAt, invalidate, isParentOf, isScrollableX, isScrollableY, layoutContainer, paint, paintIntersecting, paramString, pointerHover, pointerPressed, refreshTheme, removeAll, removeComponent, replace, replace, replaceAndWait, replaceComponents, requestFocusChild, revalidate, scrollComponentToVisible, setCellRenderer, setLayout, setScrollable, setScrollableX, setScrollableY, setShouldCalcPreferredSize

**Methods inherited from class** `com.sun.lwuit.Component`

```

addFocusListener, animate, calcPreferredSize, calcScrollSize, checkAnimation,
contains, deinitialize, deinitializeCustomStyle, deinitializeImpl, drawPainters,
fireActionEvent, fireClicked, fireFocusGained, fireFocusGained, fireFocusLost,
fireFocusLost, focusGained, focusGainedInternal, focusLost, focusLostInternal,
getAbsoluteX, getAbsoluteY, getAnimationMotion, getBaseline,
getBaselineResizeBehavior, getBorder, getBottomGap, getBounds, getClientProperty,
getComponentForm, getDirtyRegion, getHeight, getLabelForComponent, getNextFocusDown,
getNextFocusLeft, getNextFocusRight, getNextFocusUp, getParent, getPreferredH,
getPreferredSize, getPreferredW, getScrollAnimationSpeed, getScrollDimension,
getScrollX, getScrollY, getSelectCommandText, getSelectedStyle, getSideGap, getStyle,
getUIID, getUnselectedStyle, getWidth, getX, getY, handlesInput, hasFocus,
initComponent, initComponentImpl, initCustomStyle, initScrollMotion, initStyle,
isBorderPainted, isCellRenderer, isDragActivated, isEnabled, isFixedPosition,
isFocusable, isFocusPainted, isInitialized, isScrollable, isScrollableX,
isScrollableY, isScrollVisible, isSelectableInteraction, isSmoothScrolling,
isVisible, keyPressed, keyReleased, keyRepeated, laidOut, longKeyPress,
longPointerPress, mergeStyle, onEditComplete, paint, paintBackground,
paintBackgrounds, paintBorder, paintComponent, paintComponent, paintInternal,
paintInternal, paintIntersectingComponentsAbove, paintScrollbars, paintScrollbarX,
paintScrollbarY, paramString, pointerDragged, pointerDragged, pointerHover,
pointerHoverReleased, pointerPressed, pointerPressed, pointerReleased,
pointerReleased, preferredSize, putClientProperty, refreshTheme, refreshTheme,
removeFocusListener, repaint, repaint, repaint, requestFocus, scrollRectToVisible,
setAnimationMotion, setBorderPainted, setCellRenderer, setDirtyRegion,
setDragActivated, setEnabled, setFixedPosition, setFocus, setFocusable,
setFocusPainted, setHandlesInput, setHeight, setInitialized, setIsScrollVisible,
setLabelForComponent, setNextFocusDown, setNextFocusLeft, setNextFocusRight,
setNextFocusUp, setParent, setPreferredH, setPreferredSize, setPreferredW,
setScrollAnimationSpeed, setScrollSize, setScrollX, setScrollY, setSelectCommandText,
setSelectedStyle, setShouldCalcPreferredSize, setSize, setSmoothScrolling, setStyle,
setUIID, setUnselectedStyle, setVisible, setWidth, setX, setY, styleChanged, toString

```

#### Methods inherited from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait

```

#### Methods inherited from interface `com.sun.lwuit.animations.Animation`

```

animate, paint

```

#### Methods inherited from interface `com.sun.lwuit.events.StyleListener`

```

styleChanged

```

## Fields

### UV

```
private Utils.UtilsVista UV
```

Objecte per a Utilitats de generació de la Vista

## Constructors

(continued from last page)

## FormResum

```
public FormResum(Vista V,  
                 Model M,  
                 int idioma)
```

Crea una instància de FormResum

### Parameters:

V - Referència a la Vista

M - Referència al Model de dades

idioma - Identificador de l'idioma

# Vista

## Class FormSplash

```

java.lang.Object
  |
  +-com.sun.lwuit.Component
    |
    +-com.sun.lwuit.Container
      |
      +-com.sun.lwuit.Form
        |
        +-Vista.FormSplash
  
```

### All Implemented Interfaces:

com.sun.lwuit.events.StyleListener, com.sun.lwuit.animations.Animation

```

public class FormSplash
extends com.sun.lwuit.Form
  
```

Classe que defineix els elements gràfics del formulari de Splash

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

3.1 (Maig de 2010)

#### Fields inherited from class com.sun.lwuit.Form

animatableComponents, backCommand, backSK, backspaceSK, clearCommand, clearSK, commandListener, contentPane, current, cyclicFocus, defaultCommand, dragged, focusCycleRoot, focusDownSequence, focused, focusRightSequence, gameKeyListeners, glassPane, keyListeners, lastFocused, leftSK, mediaComponents, menuBar, previousForm, rightSK, rightSK2, selectCommand, tint, tintColor, title, transitionInAnimator, transitionOutAnimator

#### Fields inherited from class com.sun.lwuit.Container

cmpTransitions, components, layout, scrollableX, scrollableY, shouldLayout

#### Fields inherited from class com.sun.lwuit.Component

animationMotion, animationSpeed, beforeLastScrollX, beforeLastScrollY, BOTTOM, bounds, BRB\_CENTER\_OFFSET, BRB\_CONSTANT\_ASCENT, BRB\_CONSTANT\_DESCENT, BRB\_OTHER, CACHE\_MISS\_THREASHOLD, cellRenderer, CENTER, clientProperties, componentLabel, destScrollY, dirtyRegion, dirtyRegionLock, dragActivated, draggedMotion, enabled, fixedPosition, focusable, focused, focusListeners, focusPainted, handlesInput, id, initialized, initialScrollY, isScrollVisible, lastDragged, lastScrollX, lastScrollY, lastTime, LEFT, nextFocusDown, nextFocusLeft, nextFocusRight, nextFocusUp, painterBounds, parent, pLastDragged, preferredSize, RIGHT, scrollSize, scrollSizeRequestedByUser, scrollX, scrollY, selectedStyle, selectText, shouldCalcPreferredSize, shouldCalcScrollSize, sizeRequestedByUser, smoothScrolling, TOP, unSelectedStyle, visible

## Constructor Summary

public	<a href="#">FormSplash()</a> Crea una instància de FormSplash
--------	--

#### Methods inherited from class com.sun.lwuit.Form

actionCommand, actionCommandImpl, addCommand, addCommand, addCommandListener, addComponent, addComponent, addComponent, addComponent, addGameKeyListener, addKeyListener, addKeyListener, addSelectCommand, addSortedComponent, addSortedComponentDown, addSortedComponentRight, clearFocusVectors, createCommandList, createMenuCancelCommand, createMenuSelectCommand, createSelectCommand, deregisterAnimated, deregisterMediaComponent, dispose, disposeImpl, findAllFocusable, findScrollableAncestor, fireCommand, fireKeyEvent, focusChangeTriggersRevalidate, getBackCommand, getClearCommand, getCommand, getCommandCount, getComponentForm, getComponentIndex, getContentPane, getDefaultCommand, getFocused, getFocusPosition, getGlassPane, getLayout, getMenuStyle, getPreviousForm, getScrollAnimationSpeed, getSoftButton, getSoftButtonCount, getSoftButtonStyle, getTintColor, getTitle, getTitleComponent, getTitleStyle, getTransitionInAnimator, getTransitionOutAnimator, hasAnimations, hasMedia, hideMenu, hideNotify, hidePopups, initComponentsImpl, initFocusDown, initFocused, initFocusRight, initLaf, isCyclicFocus, isInSameColumn, isInSameRow, isMenu, isReverseSoftButtons, isSingleFocusMode, isSmoothScrolling, keyPressed, keyReleased, keyRepeated, longKeyPress, longPointerPress, onShow, onShowCompleted, paint, paintBackground, paramString, pointerDragged, pointerHover, pointerPressed, pointerReleased, refreshTheme, registerAnimated, registerMediaComponent, removeAll, removeAllCommands, removeCommand, removeCommandListener, removeComponent, removeGameKeyListener, removeKeyListener, removeKeyListener, removeSelectCommand, repaint, repaintAnimations, replace, replaceAndWait, requestFocus, restoreMenu, scrollComponentToVisible, setBackCommand, setBgImage, setClearCommand, setCommandListener, setCyclicFocus, setDefaultCommand, setDraggedComponent, setFocused, setGlassPane, setLayout, setMenuCellRenderer, setMenuStyle, setMenuTransitions, setScrollable, setScrollableX, setScrollableY, setScrollAnimationSpeed, setSmoothScrolling, setSoftButtonStyle, setTintColor, setTitle, setTitleComponent, setTitleComponent, setTitleStyle, setTransitionInAnimator, setTransitionOutAnimator, setVisible, show, show, showBack, showDialog, showMenuDialog, showModal, showModal, showNotify, showPopup, sizeChanged, sizeChangedInternal, updateFocus

#### Methods inherited from class `com.sun.lwuit.Container`

addComponent, addComponent, addComponent, addComponent, calcPreferredSize, contains, deinitializeImpl, doLayout, flushReplace, getComponentAt, getComponentAt, getComponentCount, getComponentIndex, getComponentsNames, getLayout, getLayoutHeight, getLayoutWidth, getScrollableParent, initComponentsImpl, insertComponentAt, invalidate, isParentOf, isScrollableX, isScrollableY, layoutContainer, paint, paintIntersecting, paramString, pointerHover, pointerPressed, refreshTheme, removeAll, removeComponent, replace, replace, replaceAndWait, replaceComponents, requestFocusChild, revalidate, scrollComponentToVisible, setCellRenderer, setLayout, setScrollable, setScrollableX, setScrollableY, setShouldCalcPreferredSize

#### Methods inherited from class `com.sun.lwuit.Component`

```

addFocusListener, animate, calcPreferredSize, calcScrollSize, checkAnimation,
contains, deinitialize, deinitializeCustomStyle, deinitializeImpl, drawPainters,
fireActionEvent, fireClicked, fireFocusGained, fireFocusGained, fireFocusLost,
fireFocusLost, focusGained, focusGainedInternal, focusLost, focusLostInternal,
getAbsoluteX, getAbsoluteY, getAnimationMotion, getBaseline,
getBaselineResizeBehavior, getBorder, getBottomGap, getBounds, getClientProperty,
getComponentForm, getDirtyRegion, getHeight, getLabelForComponent, getNextFocusDown,
getNextFocusLeft, getNextFocusRight, getNextFocusUp, getParent, getPreferredH,
getPreferredSize, getPreferredW, getScrollAnimationSpeed, getScrollDimension,
getScrollX, getScrollY, getSelectCommandText, getSelectedStyle, getSideGap, getStyle,
getUIID, getUnselectedStyle, getWidth, getX, getY, handlesInput, hasFocus,
initComponent, initComponentImpl, initCustomStyle, initScrollMotion, initStyle,
isBorderPainted, isCellRenderer, isDragActivated, isEnabled, isFixedPosition,
isFocusable, isFocusPainted, isInitialized, isScrollable, isScrollableX,
isScrollableY, isScrollVisible, isSelectableInteraction, isSmoothScrolling,
isVisible, keyPressed, keyReleased, keyRepeated, laidOut, longKeyPress,
longPointerPress, mergeStyle, onEditComplete, paint, paintBackground,
paintBackgrounds, paintBorder, paintComponent, paintComponent, paintInternal,
paintInternal, paintIntersectingComponentsAbove, paintScrollbars, paintScrollbarX,
paintScrollbarY, paramString, pointerDragged, pointerDragged, pointerHover,
pointerHoverReleased, pointerPressed, pointerPressed, pointerReleased,
pointerReleased, preferredSize, putClientProperty, refreshTheme, refreshTheme,
removeFocusListener, repaint, repaint, repaint, requestFocus, scrollRectToVisible,
setAnimationMotion, setBorderPainted, setCellRenderer, setDirtyRegion,
setDragActivated, setEnabled, setFixedPosition, setFocus, setFocusable,
setFocusPainted, setHandlesInput, setHeight, setInitialized, setIsScrollVisible,
setLabelForComponent, setNextFocusDown, setNextFocusLeft, setNextFocusRight,
setNextFocusUp, setParent, setPreferredH, setPreferredSize, setPreferredW,
setScrollAnimationSpeed, setScrollSize, setScrollX, setScrollY, setSelectCommandText,
setSelectedStyle, setShouldCalcPreferredSize, setSize, setSmoothScrolling, setStyle,
setUIID, setUnselectedStyle, setVisible, setWidth, setX, setY, styleChanged, toString

```

#### Methods inherited from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait

```

#### Methods inherited from interface `com.sun.lwuit.animations.Animation`

```

animate, paint

```

#### Methods inherited from interface `com.sun.lwuit.events.StyleListener`

```

styleChanged

```

## Constructors

### FormSplash

```
public FormSplash()
```

Crea una instància de FormSplash

# Vista

## Class Vista

```
java.lang.Object
```

```
└--Vista.Vista
```

### All Implemented Interfaces:

```
com.sun.lwuit.events.ActionListener
```

```
public class Vista
extends java.lang.Object
implements com.sun.lwuit.events.ActionListener
```

Classe que conté tota la gestió de la GUI (Graphic User Interface). Actua com a Vista dins del patró MVC (Model-View-Controller)

### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

### Version:

3.1 (Maig de 2010)

## Nested Class Summary

class	<a href="#">Vista.endSplashScreen</a> Vista.endSplashScreen
class	<a href="#">Vista.endTimerRefresc</a> Vista.endTimerRefresc

## Field Summary

private	<a href="#">CB_NovaSensibilitat</a> ComboBox del Dialog per a selecció de la nova sensibilitat a TX
private	<a href="#">cmd1</a> Command Button esquerre del Dialog
private	<a href="#">cmd2</a> Command Button dret del Dialog
private	<a href="#">D</a> Objecte Dialog
private	<a href="#">formActual</a> Identificador del formulari que s'està mostrant
private	<a href="#">formAnterior</a> Identificador del formulari mostrat anteriorment al que s'està mostrant
private	<a href="#">M</a> Referència al Model de dades
private	<a href="#">P</a> Referència al MIDlet



private	<a href="#"><u>T</u></a> Timer par a timeouts
private	<a href="#"><u>UV</u></a> Objecte per a Utilitats de generació de la Vista

## Constructor Summary

public	<a href="#"><u>Vista</u></a> ( <a href="#"><u>PPStop</u></a> P, <a href="#"><u>Model</u></a> M) Crea una instància de Vista
--------	--

## Method Summary

void	<a href="#"><u>actionPerformed</u></a> (com.sun.lwuit.events.ActionEvent evt) Listener d'events de Command Button: Realitza es accions pertinents
void	<a href="#"><u>initComponents</u></a> () Inicialitza la Graphic User Interface (GUI)
void	<a href="#"><u>mostrarDialog</u></a> (int tipus, java.lang.String text, com.sun.lwuit.Command cmd1, com.sun.lwuit.Command cmd2, long timeout, int height) Mostra per pantalla un Dialog d'un cert tipus i amb una certa configuració
void	<a href="#"><u>mostrarDialogSensibilitat</u></a> (com.sun.lwuit.Command cmd1, com.sun.lwuit.Command cmd2) Mostra per pantalla el Dialog de Nova Sensibilitat
void	<a href="#"><u>mostrarFormulari</u></a> (int f) Mostra per pantalla un determinat formulari
void	<a href="#"><u>setDefaultStyles</u></a> (java.lang.String recurs, java.lang.String tema) Defineix els estils per defecte dels components de la vista: carrega un tema d'un determinat recurs
void	<a href="#"><u>update</u></a> () Actualitza la Vista

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Methods inherited from interface com.sun.lwuit.events.ActionListener

actionPerformed

## Fields

### P

private Main.PPStop **P**

Referència al MIDlet

(continued from last page)

---

## M

private Dades.Model **M**

Referència al Model de dades

---

## T

private java.util.Timer **T**

Timer par a timeouts

---

## formActual

private int **formActual**

Identificador del formulari que s'està mostrant

---

## formAnterior

private int **formAnterior**

Identificador del formulari mostrat anteriorment al que s'està mostrant

---

## D

private com.sun.lwuit.Dialog **D**

Objecte Dialog

---

## cmd1

private com.sun.lwuit.Command **cmd1**

Command Button esquerre del Dialog

---

## cmd2

private com.sun.lwuit.Command **cmd2**

Command Button dret del Dialog

---

## CB\_NovaSensibilitat

private com.sun.lwuit.ComboBox **CB\_NovaSensibilitat**

ComboBox del Dialog per a selecció de la nova sensibilitat a TX

---

## UV

private Utils.UtilsVista **UV**

Objecte per a Utilitats de generació de la Vista

## Constructors

---

## Vista

```
public Vista(PPStop P,  
             Model M)
```

Crea una instància de Vista

### Parameters:

- P - Referència al MIDlet
- M - Referència al Model de dades

## Methods

### mostrarFormulari

```
public void mostrarFormulari(int f)
```

Mostra per pantalla un determinat formulari

### Parameters:

- f - Identificador del formulari a mostrar

### setDefaultStyles

```
private void setDefaultStyles(java.lang.String recurs,  
                               java.lang.String tema)
```

Defineix els estils per defecte dels components de la vista: carrega un tema d'un determinat recurs

### Parameters:

- recurs - Recurs en el que es troba el tema a aplicar
- tema - Tema a aplicar

### mostrarDialog

```
public void mostrarDialog(int tipus,  
                           java.lang.String text,  
                           com.sun.lwuit.Command cmd1,  
                           com.sun.lwuit.Command cmd2,  
                           long timeout,  
                           int height)
```

Mostra per pantalla un Dialog d'un cert tipus i amb una certa configuració

### Parameters:

- tipus - Tipus de Dialog
- text - Text que ha de mostrar el Dialog
- cmd1 - Command Button de l'esquerre del Dialog
- cmd2 - Command Button de la dreta del Dialog
- timeout - Timeout del Dialog (ms)
- height - Alçada del Dialog (px)

### mostrarDialogSensibilitat

```
public void mostrarDialogSensibilitat(com.sun.lwuit.Command cmd1,  
                                       com.sun.lwuit.Command cmd2)
```

Mostra per pantalla el Dialog de Nova Sensibilitat

(continued from last page)

**Parameters:**

cmd1 - Command Button de l'esquerra del Dialog  
cmd2 - Command Button de la dreta del Dialog

---

**initComponents**

```
private void initComponents()
```

Inicialitza la Graphic User Interface (GUI)

---

**update**

```
public void update()
```

Actualitza la Vista

---

**actionPerformed**

```
public void actionPerformed(com.sun.lwuit.events.ActionEvent evt)
```

Listener d'events de Command Button: Realitza es accions pertinents

**Parameters:**

evt - Event generat

## Vista

### Class Vista.endSplashScreen

```

java.lang.Object
  |
  +- java.util.TimerTask
        +- Vista.Vista.endSplashScreen

```

#### All Implemented Interfaces:

```
java.lang.Runnable
```

```

private class Vista.endSplashScreen
extends java.util.TimerTask

```

Final del Timer de Splash Screen: Mostrar Men Principal

#### Fields inherited from class java.util.TimerTask

CANCELLED, EXECUTED, lock, nextExecutionTime, period, SCHEDULED, state, VIRGIN

### Constructor Summary

private	<a href="#">Vista.endSplashScreen()</a>
---------	---

### Method Summary

void	<a href="#">run()</a>
------	-----------------------

#### Methods inherited from class java.util.TimerTask

cancel, run, scheduledExecutionTime

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Runnable

run

### Constructors

#### Vista.endSplashScreen

```
private Vista.endSplashScreen()
```

### Methods

(continued from last page)

## **run**

```
public void run()
```

# Vista

## Class Vista.endTimerRefresc

```

java.lang.Object
  |
  +- java.util.TimerTask
        |
        +- Vista.Vista.endTimerRefresc
  
```

### All Implemented Interfaces:

```
java.lang.Runnable
```

```

private class Vista.endTimerRefresc
extends java.util.TimerTask
  
```

Final del Timer de Refresc: Refrescar formulari

### Fields inherited from class java.util.TimerTask

CANCELLED, EXECUTED, lock, nextExecutionTime, period, SCHEDULED, state, VIRGIN

## Constructor Summary

private	<a href="#">Vista.endTimerRefresc()</a>
---------	---

## Method Summary

void	<a href="#">run()</a>
------	-----------------------

### Methods inherited from class java.util.TimerTask

cancel, run, scheduledExecutionTime

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Methods inherited from interface java.lang.Runnable

run

## Constructors

### Vista.endTimerRefresc

```
private Vista.endTimerRefresc()
```

## Methods

(continued from last page)

**run**

```
public void run()
```



## C.3 Codi de l'aplicació

Codi C.1: (AppControl) Alarma.java

```
1 package Audio;
2
3 import Constants.CONST;
4 import Dades.Model;
5
6 /**
7  * Classe que crea un player per a la reproducció de l'Alarma PPStop
8  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
9  * @version 3.1 (Maig de 2010)
10 */
11 public class Alarma {
12
13     /** Referència al Model de dades */
14     private Model M;
15     /** Objecte d'Audio */
16     private Audio A;
17
18     /**
19      * Crea una instància de Alarma
20      * @param M      Model de dades
21      */
22     public Alarma(Model M){
23         this.M = M;
24         update();
25     }
26
27     /**
28      * Activa l'alarma sonora
29      */
30     public void playAlarma(){
31         A.playAudio();
32     }
33
34     /**
35      * Atura l'alarma sonora
36      */
37     public void stopAlarma(){
38         A.stopAudio();
39     }
40
41     /**
42      * Ajusta el volum de l'alarma
43      * @param nivell  Nivell de volum (%)
44      */
45     public void setVolum(int nivell){
46         A.setVolum(nivell);
47     }
48
49     /**
50      * Actualitza el player amb la nova informació del Model de Dades
51      */
52     public void update(){
53         if (M.getTipusAlarma() != CONST.ALARMA_SILENCI){
54             A = new Audio(CONST.ALARMA_ARXIUS[M.getTipusAlarma()], "audio/amr", M.
55                 getVolumAlarma(), -1, false);
56             A.start();
57         }
58     }
59 }
```

Codi C.2: (AppControl) Audio.java

```
1 package Audio;
```

```

2
3 import java.io.IOException;
4 import java.io.InputStream;
5 import javax.microedition.media.Manager;
6 import javax.microedition.media.MediaException;
7 import javax.microedition.media.Player;
8 import javax.microedition.media.control.VolumeControl;
9
10 /**
11  * Classe que crea un player per a la reproducció d'arxius d'àudio
12  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
13  * @version 3.1 (Maig de 2010)
14  */
15 public class Audio extends Thread {
16
17     /** Objecte Player */
18     private Player PL;
19     /** Arxiu a reproduir */
20     private String arxiu;
21     /** Tipus d'arxiu d'àudio a reproduir */
22     private String tipus;
23     /** Volum de reproducció (%) */
24     private int volum;
25     /** Nombre de repeticions de reproducció */
26     private int loops;
27     /** Començar automàticament l'àudio en crear l'objecte */
28     private boolean autostart;
29
30     /**
31      * Crea una instància de Audio
32      * @param arxiu      Arxiu a reproduir
33      * @param tipus      Tipus d'arxiu d'àudio a reproduir
34      * @param loops      Començar automàticament l'àudio en crear l'objecte
35      */
36     public Audio(String arxiu, String tipus, int volum, int loops, boolean autostart){
37         this.arxiu = arxiu;
38         this.tipus = tipus;
39         this.volum = volum;
40         this.loops = loops;
41         this.autostart = autostart;
42     }
43
44     /**
45      * Crea un player per a la reproducció de l'àudio
46      */
47     private void crearPlayer(){
48         InputStream in = getClass().getResourceAsStream(this.arxiu);
49         try {
50             PL = Manager.createPlayer(in, this.tipus);
51             PL.realize();
52             PL.prefetch();
53         } catch (IOException ex) {
54             ex.printStackTrace();
55         } catch (MediaException ex) {
56             ex.printStackTrace();
57         }
58         PL.setLoopCount(this.loops);
59         setVolum(volum);
60     }
61
62     /**
63      * Inicia la reproducció de l'àudio
64      */
65     public void playAudio(){
66         try {
67             PL.start();
68         } catch (MediaException ex) {
69             ex.printStackTrace();
70         }
71     }

```

```

72
73  /**
74   * Atura la reproducció de l'àudio
75   */
76  public void stopAudio(){
77      try {
78          PL.stop();
79          PL.setMediaTime(0);
80      } catch (MediaException ex) {
81          ex.printStackTrace();
82      }
83  }
84
85  /**
86   * Ajusta el volum de l'àudio
87   * @param nivell    Volum de reproducció (%)
88   */
89  public void setVolum(int nivell){
90      VolumeControl VC = (VolumeControl)PL.getControl("VolumeControl");
91      VC.setLevel(nivell);
92  }
93
94  /**
95   * Crea un player per a la reproducció d'àudio i, si escau, el reprodueix
96   */
97  public void run(){
98      crearPlayer();
99      if(this.autostart)
100          playAudio();
101  }
102
103 }

```

### Codi C.3: (AppControl) Connexio.java

```

1 package Bluetooth;
2
3 import Dades.Historial;
4 import java.io.DataInputStream;
5 import java.io.DataOutputStream;
6 import java.io.IOException;
7 import javax.microedition.io.StreamConnection;
8
9 /**
10  * Classe que obre els streams d'e I/O per a la transmissió i recepció de dades
11  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
12  * @version 3.1 (Maig de 2010)
13  */
14 public class Connexio {
15
16     /** Stream d'entrada de dades */
17     private DataInputStream dis;
18     /** Stream de sortida de dades */
19     private DataOutputStream dos;
20
21     /** Crea una nova instància de Connexio */
22     public Connexio(){}
23
24     /**
25      * Obté els Streams de I/O a partir d'un l'StreamConnection
26      * @param SC    Stream de la connexió
27      */
28     public void obrirDataStreams(StreamConnection SC){
29         try {
30             dis = SC.openDataInputStream();
31             dos = SC.openDataOutputStream();
32         } catch (IOException ex) {
33             ex.printStackTrace();
34         }
35     }
36 }

```

```

35     }
36
37     /**
38      * Tanca els Streams de I/O
39      */
40     public void tancarDataStreams(){
41         try {
42             dis.close();
43             dos.close();
44         } catch (IOException ex) {
45             ex.printStackTrace();
46         }
47     }
48
49     /**
50      * Transmet un Missatge a través de l'OutputStream
51      * @param msg    Missatge a enviar
52      */
53     public void enviarMSG(String msg){
54         try {
55             dos.write(msg.getBytes());
56             dos.flush();
57         } catch (IOException ex) {
58             ex.printStackTrace();
59         }
60     }
61
62     /**
63      * Rep un Missatge a través de l'InputStream
64      * @return Missatge rebut
65      */
66     public String rebreMSG(){
67         int num_bytes;
68         String msg = null;
69         try {
70             byte[] buffer = new byte[10];
71             num_bytes = dis.read(buffer);
72             msg = new String(buffer,0,num_bytes);
73         } catch (IOException ex) {
74             ex.printStackTrace();
75         }
76         return msg;
77     }
78
79     /**
80      * Transmet l'Historial a través de l'OutputStream
81      * @param H        Historial a transmetre
82      */
83     public void enviarHIST(Historial H){
84         try {
85             dos.write(H.toByteArray());
86             dos.flush();
87         } catch (IOException ex) {
88             ex.printStackTrace();
89         }
90     }
91
92 }

```

#### Codi C.4: (AppControl) Servei.java

```

1 package Bluetooth;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import Dades.Model;
6 import Main.PPStop;
7 import com.sun.lwuit.Dialog;
8 import java.io.*;

```

```

9 import javax.bluetooth.*;
10 import javax.microedition.io.*;
11
12 /**
13  * Classe que crea registra el servei bluetooth PPStop i tracta les connexions rebudes
14  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
15  * @version 3.1 (Maig de 2010)
16  */
17 public class Servei extends Thread {
18
19     /** Referència al MIDlet */
20     private PPStop P;
21     /** Referència al Model de dades */
22     private Model M;
23     /** Objecte de Connexió */
24     private Connexio C = new Connexio();
25     /** Missatge a enviar */
26     private String msg;
27     /** Adreça bluetooth origen de la connexió */
28     private String addrOrigen;
29     /** Flag a l'inici de la connexió */
30     private int flagPrevi;
31
32     /**
33      * Crea una instància de Servei
34      * @param P      Referència al MIDlet
35      * @param M      Referència al Model de dades
36      */
37     public Servei(PPStop P, Model M){
38         this.P = P;
39         this.M = M;
40     }
41
42     /**
43      * Registra el servei bluetooth PPStop
44      */
45     private void crearServei(){
46         try {
47             LocalDevice LD = LocalDevice.getLocalDevice();
48             try {
49                 LocalDevice.getLocalDevice().getFriendlyName();
50                 M.setEstatBT(CONST.ESTAT_ACTIVAT);
51                 M.setMacMovel(LD.getBluetoothAddress());
52             } catch (BluetoothStateException e) {
53                 M.setEstatBT(CONST.ESTAT_DESACTIVAT);
54             }
55             DiscoveryAgent DA = LD.getDiscoveryAgent();
56             LD.setDiscoverable(DA.GIAC);
57             String URL = "btspp://localhost:" + CONST.SERVEI_UUID + ";name=" + CONST.
58                 SERVEI_NOM;
59             StreamConnectionNotifier SRV = (StreamConnectionNotifier) Connector.open(URL
60                 );
61             StreamConnection SC;
62             while (true){
63                 SC = (StreamConnection) SRV.acceptAndOpen();
64                 flagPrevi = M.getFlag();
65                 tractarConnexio(SC);
66                 SC.close();
67             }
68         } catch (IOException ex) {
69             ex.printStackTrace();
70         }
71     }
72
73     /**
74      * Tracta la connexió bluetooth rebuda pel servei PPStop
75      * @param SC      Stream de la connexió a tractar
76      */
77     private void tractarConnexio(StreamConnection SC){
78         C.obrirDataStreams(SC);
79     }
80 }

```

```

77     msg = C.rebreMSG();
78     RemoteDevice RD = null;
79     try {
80         RD = RemoteDevice.getRemoteDevice(SC);
81     } catch (IOException ex) {
82         ex.printStackTrace();
83     }
84     addrOrigen = RD.getBluetoothAddress();
85     String ordre = tractarMSG(msg, CONST.OBTENIR_ORDRE);
86     if (addrOrigen.equalsIgnoreCase(M.getCodi())){
87         if (ordre.equals(CONST.ORDRE_COMPROVACIO)){
88             C.enviarMSG("OK");
89             M.setEstat(CONST.ESTAT_ASSOCIAT);
90             actualitzaParametres(msg);
91             P.mostrarDialogInformatiu(Dialog.TYPE_INFO, LANG.TEXT_DISPASSOCIAT[M.
                getIdioma()], 0, 50);
92         } else if (ordre.equals(CONST.ORDRE_INFODISPOSITIU)){
93             actualitzaParametres(msg);
94         } else if (ordre.equals(CONST.ORDRE_ALERTAR)){
95             switch (M.getFlag()){
96                 case CONST.FLAG_ALARMA:
97                     P.StartAlarma();
98                     C.enviarMSG("OK");
99                     M.setEstat(CONST.ESTAT_ASSOCIAT);
100                    actualitzaParametres(msg);
101                    break;
102                 case CONST.FLAG_BUSCARTERMINALS:
103                     C.enviarMSG("B");
104                     M.setFlag(CONST.FLAG_ALARMA);
105                     M.setEstat(CONST.ESTAT_NOASSOCIAT);
106                     resetParametres();
107                     P.mostrarDialogInformatiu(Dialog.TYPE_INFO, LANG.
                        TEXT_BUSQUEDAPROGRES[M.getIdioma()], 0, 50);
108                     break;
109                 case CONST.FLAG_MODIFICARSENSIBILITAT:
110                     C.enviarMSG("S" + String.valueOf(M.getSensibilitatTX()));
111                     M.setSensibilitat(M.getSensibilitatTX());
112                     M.setFlag(CONST.FLAG_ALARMA);
113                     actualitzaParametres(msg);
114                     P.mostrarDialogInformatiu(Dialog.TYPE_INFO, LANG.
                        TEXT_SENSCANVIADA[M.getIdioma()], 0, 50);
115                     break;
116                 case CONST.FLAG_ROTARTERMINALS:
117                     C.enviarMSG("R");
118                     M.setFlag(CONST.FLAG_ALARMA);
119                     P.mostrarDialogInformatiu(Dialog.TYPE_INFO, LANG.
                        TEXT_TERMINALROTAT[M.getIdioma()], 0, 50);
120                     break;
121             }
122         }
123     }
124     if (ordre.equals(CONST.ORDRE_TRANSFERIRHISTORIAL)) {
125         C.enviarHIST(M.getHistorial());
126         P.mostrarDialogInformatiu(Dialog.TYPE_INFO, LANG.TEXT_HISTTRANSFERIT[M.
            getIdioma()], 0, 50);
127     }
128     C.tancarDataStreams();
129 }
130
131 /**
132  * Tracta un misatge rebut i n'extreu el paràmetre indicat (ORDRE, BATERIA,
133  * SENSIBILITAT)
134  * @param msg Missatge a tractar
135  * @param tipus Tipus de dada a obtenir
136  * @return Paràmetre sol·licitat
137  */
138 private String tractarMSG(String msg, int tipus){
139     switch(tipus){
140         case CONST.OBTENIR_ORDRE: return msg.substring(0,1);
141         case CONST.OBTENIR_BATERIA: return msg.substring(1,4);

```

```

141         case CONST.OBTENIR_SENSIBILITAT: return msg.substring(4,5);
142         case CONST.OBTENIR_ORDRETERMINAL: return msg.substring(5,6);
143         default: return null;
144     }
145 }
146
147 /**
148  * Actualitza en el model la sensibilitat, estat de bateria del dispositiu i ordre
149  * del dispositiu
150  * @param msg Missatge que conté els paràmetres a actualitzar
151  */
152 private void actualitzaParametres(String msg){
153     M.setBateria(Integer.parseInt(tractarMSG(msg,CONST.OBTENIR_BATERIA)));
154     if (flagPrevi != CONST.FLAG_MODIFICARSENSIBILITAT)
155         M.setSensibilitat(Integer.parseInt(tractarMSG(msg,CONST.OBTENIR_SENSIBILITAT)));
156     M.setOrdre(Integer.parseInt(tractarMSG(msg,CONST.OBTENIR_ORDRETERMINAL)));
157 }
158
159 /**
160  * Reseteja en el model la sensibilitat, estat de bateria del dispositiu i ordre del
161  * dispositiu
162  */
163 private void resetParametres(){
164     M.setBateria(CONST.NODEF);
165     M.setSensibilitat(CONST.NODEF);
166     M.setOrdre(CONST.NODEF);
167 }
168
169 /**
170  * Crea el servei bluetooth PPStop i tracta les connexions rebudes
171  */
172 public void run(){
173     crearServei();
174 }
175 }

```

### Codi C.5: (AppControl) CONST.java

```

1 package Constants;
2
3 import javax.bluetooth.UUID;
4
5 /**
6  * Interfície que conté totes les constants numèriques utilitzades per l'aplicació
7  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
8  * @version 3.1 (Maig de 2010)
9  */
10 public interface CONST {
11
12     /** Temps de l'Splash Screen (ms) */
13     public static final int TEMPS_SPLASHSCREEN = 3000;
14     /** Temps de Dialog informatiu (ms) */
15     public static final int TEMPS_DIALOG = 3000;
16     /** Temps de Dialog informatiu (ms) de búsqueda en progres... */
17     public static final int TEMPS_DIALOG_BUSQUEDA = 15000;
18
19     /** Formulari de l'Splash Screen */
20     public static final int FORM_SPLASH = 1;
21     /** Formulari del Men Principal */
22     public static final int FORM_MENU = 2;
23     /** Formulari de Resum de Dades */
24     public static final int FORM_RESUM = 3;
25     /** Formulari de Buscar Terminals */
26     public static final int FORM_BUSCARTERMINALS = 4;
27     /** Formulari de Modificar Sensibilitat */
28     public static final int FORM_MODIFICARSENSIBILITAT = 5;
29     /** Formulari de configuració de l'aplicació */
30     public static final int FORM_CONFIGURACIO = 6;

```

```

31  /** Formulari de l'Historial d'Alarmes */
32  public static final int FORM_HISTORIALALARMES = 7;
33  /** Formulari de Transferència de l'Historial */
34  public static final int FORM_TRANSFERIRHISTORIAL = 8;
35  /** Formulari de l'Alarma */
36  public static final int FORM_ALARMA = 9;
37  /** Formulari anterior al mostrat actualment */
38  public static final int FORM_ANTERIOR = 10;
39  /** Formulari de rotació de terminals */
40  public static final int FORM_ROTARTERMINALS = 11;
41
42  /** Command de no dur a terme cap acció */
43  public static final int COMMAND_RES = 1;
44  /** Command per a confirmació de sortir de l'aplicació */
45  public static final int COMMAND_SORTIRDIALOG = 2;
46  /** Command per a sortir de l'aplicació */
47  public static final int COMMAND_SORTIR = 3;
48  /** Command per a aturar l'alarma */
49  public static final int COMMAND_ATURAR = 4;
50  /** Command per a tornar al formulari anterior */
51  public static final int COMMAND_TORNAR = 5;
52  /** Command per a confirmació d'esborrar l'historial */
53  public static final int COMMAND_ESBORRARDIALOG = 6;
54  /** Command per a esborrar l'historial */
55  public static final int COMMAND_ESBORRAR = 7;
56  /** Command per a canviar el flag a Buscar Terminals */
57  public static final int COMMAND_BUSCARTERMINALS = 8;
58  /** Command per a canviar el flag a Modificar Sensibilitat */
59  public static final int COMMAND_MODIFICARSENSIBILITAT = 9;
60  /** Command per a canviar el flag al valor per defecte */
61  public static final int COMMAND_CANCELARACCIO = 10;
62  /** Command per a canviar el flag de Rotar Terminals */
63  public static final int COMMAND_ROTARTERMINALS = 11;
64
65  /** Representa un valor no definit */
66  public static final int NODEF = -1;
67
68  /** El terminal es troba associat a un dispositiu PPStop */
69  public static final int ESTAT_ASSOCIAT = 1;
70  /** El terminal NO es troba associat a cap dispositiu PPStop */
71  public static final int ESTAT_NOASSOCIAT = 2;
72
73  /** El terminal té la connectivitat bluetooth activada */
74  public static final int ESTAT_ACTIVAT = 1;
75  /** El terminal té la connectivitat bluetooth desactivada */
76  public static final int ESTAT_DESACTIVAT = 2;
77
78  /** La propera acció a realitzar, donada una connexió PPStop, és la d'alarmar */
79  public static final int FLAG_ALARMA = 1;
80  /** La propera acció a realitzar, donada una connexió PPStop, és la de buscar
    terminals */
81  public static final int FLAG_BUSCARTERMINALS = 2;
82  /** La propera acció a realitzar, donada una connexió PPStop, és la de modificar la
    sensibilitat */
83  public static final int FLAG_MODIFICARSENSIBILITAT = 3;
84  /** La propera acció a realitzar, donada una connexió PPStop, és la de rotar
    cíclicament la posició dels terminals */
85  public static final int FLAG_ROTARTERMINALS = 4;
86
87  /** L'idioma de la interfície de l'aplicació és el Català */
88  public static final int IDIOMA_CA = 0;
89  /** L'idioma de la interfície de l'aplicació és el Castellà */
90  public static final int IDIOMA_ES = 1;
91  /** L'idioma de la interfície de l'aplicació és l'Anglès */
92  public static final int IDIOMA_EN = 2;
93
94  /** El tipus d'alarma sonora és una veu humana */
95  public static final int ALARMA_VEUHUMANA = 0;
96  /** El tipus d'alarma sonora és un RING convencional */
97  public static final int ALARMA_RING = 1;

```



```

98  /** No s'activa cap tipus d'alarma sonora */
99  public static final int ALARMA_SILENCI = 2;
100
101  /** L'alarma es desactiva automàticament passats 10 segons */
102  public static final int ALARMA_T10S = 10000;
103  /** L'alarma es desactiva automàticament passats 20 segons */
104  public static final int ALARMA_T20S = 20000;
105  /** L'alarma es desactiva automàticament passats 30 segons */
106  public static final int ALARMA_T30S = 30000;
107  /** L'alarma mai es desactiva automàticament */
108  public static final int ALARMA_TINDEFINIT = 0;
109
110  /** El volum de l'alarma és del 20% */
111  public static final int ALARMA_VOL20 = 20;
112  /** El volum de l'alarma és del 40% */
113  public static final int ALARMA_VOL40 = 40;
114  /** El volum de l'alarma és del 60% */
115  public static final int ALARMA_VOL60 = 60;
116  /** El volum de l'alarma és del 80% */
117  public static final int ALARMA_VOL80 = 80;
118  /** El volum de l'alarma és del 100% */
119  public static final int ALARMA_VOL100 = 100;
120
121  /** Arxius d'àudio d'alarma: Veu Humana / RING */
122  public static final String[] ALARMA_ARXIUS = {"alarma_veu.amr", "alarma_ring.amr"};
123
124  /** Data en format curt */
125  public static final int DATA_CURTA = 1;
126  /** Data en format llarg */
127  public static final int DATA_LLARGA = 2;
128
129  /** Obtenir el paràmetre ORDRE en el MSG rebut */
130  public static final int OBTENIR_ORDRE = 1;
131  /** Obtenir el paràmetre BATERIA en el MSG rebut */
132  public static final int OBTENIR_BATERIA = 2;
133  /** Obtenir el paràmetre BATERIA en el MSG rebut */
134  public static final int OBTENIR_SENSIBILITAT = 3;
135  /** Obtenir el paràmetre ORDRE DEL TERMINAL en el MSG rebut */
136  public static final int OBTENIR_ORDRETERMINAL = 4;
137
138  /** L'ordre rebuda és comprovar codi PPStop: associar servei PPStop */
139  public static final String ORDRE_COMPROVACIO = "C";
140  /** L'ordre rebuda és la d'alertar (alarma PPStop) */
141  public static final String ORDRE_ALERTAR = "A";
142  /** L'ordre rebuda és la de transferir historial */
143  public static final String ORDRE_TRANSFERIRHISTORIAL = "T";
144  /** L'ordre rebuda és la d'actualitzar la informació del dispositiu PPStop */
145  public static final String ORDRE_INFODISPOSITIU = "I";
146
147  /** RecordSet de la Configuració */
148  public static final String RS_CONFIG = "PPStop_config";
149  /** RecordSet de l'Historial */
150  public static final String RS_HISTORIAL = "PPStop_historial";
151
152  /** Nom del servei PPStop a registrar */
153  public static final String SERVEI_NOM = "PPStop";
154  /** UUID del servei PPStop a registrar */
155  public static final String SERVEI_UUID = new UUID(0x1101).toString();
156
157  /** Representa un valor buit */
158  public static final String EMPTY = "";
159
160 }

```

## Codi C.6: (AppControl) LANG.java

```

1 package Constants;
2
3 /**

```

```

4  * Interfcie que conté totes les constants d'idioma utilitzades per l'aplicació
5  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
6  * @version 3.1 (Maig de 2010)
7  */
8  public interface LANG {
9
10     /** Títol del formulari Principal */
11     public static final String[] TITOL_PRINCIPAL = {"PPStop","PPStop","PPStop"};
12     /** Títol del formulari Resum de Dades */
13     public static final String[] TITOL_RESUM = {"Resum","Resumen","Summary"};
14     /** Títol del formulari de Configuració de l'Aplicació */
15     public static final String[] TITOL_CONFIGURACIO = {"Configuració","Configuración","Settings"};
16     /** Títol del formulari de Buscar Terminals */
17     public static final String[] TITOL_BUSCARTERMINALS = {"Buscar Terminals","Buscar Terminales","Find Devices"};
18     /** Títol del formulari de Modificar Sensibilitat */
19     public static final String[] TITOL_MODIFICARSENSIBILITAT = {"Modificar Sensibilitat","Modificar Sensibilidad","Modify Sensibility"};
20     /** Títol del formulari d'Historial d'Alarmes */
21     public static final String[] TITOL_HISTORIALALARMES = {"Historial d'Alarmes","Historial de Alarmas","Alarm History"};
22     /** Títol del formulari de Rotar Terminals */
23     public static final String[] TITOL_ROTARTERMINALS = {"Rotar Terminals","Rotar Terminales","Rotate devices"};
24     /** Títol del formulari de Transferir Historial */
25     public static final String[] TITOL_TRANSFERIRHISTORIAL = {"Transferir Historial","Transferir Historial","Transfer History"};
26     /** Títol del formulari d'Alarma */
27     public static final String[] TITOL_ALARMA = {"Alarma","Alarma","Alarm"};
28
29     /** Label del command d'aturar alarma */
30     public static final String[] MENU_ATURAR = {"Aturar","Parar","Stop"};
31     /** Label del command de tornar al formulari anterior */
32     public static final String[] MENU_TORNAR = {"Tornar","Volver","Go back"};
33     /** Label del command de sortir de l'aplicació */
34     public static final String[] MENU_SORTIR = {"Sortir","Salir","Exit"};
35     /** Label del command d'esborrar historial */
36     public static final String[] MENU_ESBORRAR = {"Esborrar","Borrar","Delete"};
37     /** Label del command de confirmació SI */
38     public static final String[] MENU_SI = {"Si","Si","Yes"};
39     /** Label del command de confirmació NO */
40     public static final String[] MENU_NO = {"No","No","No"};
41     /** Label del command de tancar Dialog Informatiu no finit */
42     public static final String[] MENU_OK = {"OK","OK","OK"};
43
44     /** Label del formulari resum: Estat Bluetooth */
45     public static final String[] LABEL_ESTATBLUETOOTH = {"Estat Bluetooth:","Estado Bluetooth:","Bluetooth Status:"};
46     /** Label del formulari resum: Codi PPStop */
47     public static final String[] LABEL_CODIPPSTOP = {"Codi PPStop:","Código PPStop:","PPStop Code:"};
48     /** Label del formulari resum: Codi Historial */
49     public static final String[] LABEL_CODIHISTORIAL = {"Codi Historial:","Código Historial:","History Code:"};
50     /** Label del formulari resum: Estat Servei */
51     public static final String[] LABEL_ESTATSERVEI = {"Estat Servei:","Estado Servicio:","Service Status:"};
52     /** Label del formulari resum: Ordre Terminal */
53     public static final String[] LABEL_ORDRETERMINAL = {"Ordre Terminal:","Orden Terminal:","Device Order:"};
54     /** Label del formulari resum: Estat Bateria */
55     public static final String[] LABEL_ESTATBATERIA = {"Estat Bateria:","Estado Batería:","Battery Status:"};
56     /** Label del formulari resum: Sensibilitat */
57     public static final String[] LABEL_SENSIBILITAT = {"Sensibilitat:","Sensibilidad:","Sensitivity:"};
58     /** Label del formulari resum: Darrera Alarma */
59     public static final String[] LABEL_DARRERAALARMA = {"Darrera Alarma:","Última Alarma:","Last Alarm"};

```

```

60  /** Label del formulari resum: Pròxima Acció */
61  public static final String[] LABEL_PROXIMAACCIO = {"Pròxima Acció:", "Próxima Acción",
62  , "Next Behaviour"};
63  /** Label del formulari resum: (baixa) */
64  public static final String[] LABEL_BAIXA = {"baixa", "baja", "low"};
65  /** Label del formulari resum: (alta) */
66  public static final String[] LABEL_ALTA = {"alta", "alta", "high"};
67  /** Label del formulari resum: NO ASSOCIAT */
68  public static final String[] LABEL_NOASSOCIAT = {"NO ASSOCIAT", "NO ASOCIADO", "NO
69  MATCHED"};
70  /** Label del formulari resum: ASSOCIAT */
71  public static final String[] LABEL_ASSOCIAT = {"ASSOCIAT", "ASOCIADO", "MATCHED"};
72  /** Label del formulari resum: ACTIU */
73  public static final String[] LABEL_ACTIVAT = {"ACTIU", "ACTIVO", "ACTIVE"};
74  /** Label del formulari resum: INACTIU */
75  public static final String[] LABEL_DESACTIVAT = {"INACTIU", "INACTIVO", "INACTIVE"};
76  /** Label del formulari resum: Comprovar */
77  public static final String[] TEXT_FLAGCOMPROVAR = {"Comprovar", "Comprobar", "Check"};
78  /** Label del formulari resum: Alarmar */
79  public static final String[] TEXT_FLAGALARMA = {"Alarmar", "Alarmar", "Alarm"};
80  /** Label del formulari resum: Buscar terminals */
81  public static final String[] TEXT_FLAGBUSCARTERMINALS = {"Buscar terminals", "Buscar
82  terminales", "Find devices"};
83  /** Label del formulari resum: Canvi sen. */
84  public static final String[] TEXT_FLAGCANVIARSENSIBILITAT = {"Canvi sen.", "Cambio
85  sen.", "Change sen."};
86  /** Label del formulari resum: Rotar terminals */
87  public static final String[] TEXT_FLAGROTARTERMINALS = {"Rotar terminals", "Rotar
88  terminales", "Rotate devices"};
89  /** Label del formulari resum: Fa < 1 minut */
90  public static final String[] TEXT_FAMENYSUNMINUT = {"Fa < 1 minut", "Hace < 1 minute
91  ", "< 1 min ago"};
92  /** Label del formulari resum: Fa 1 minut */
93  public static final String[] TEXT_FA1MINUT = {"Fa 1 minut", "Hace 1 minuto", "1 minute
94  ago"};
95  /** Label del formulari resum: Fa {X} mins */
96  public static final String[] TEXT_FAXMINUTS = {"Fa {X} mins", "Hace {X} mins", "{X}
97  mins ago"};
98  /** Label del formulari resum: Fa 1 hora */
99  public static final String[] TEXT_FA1HORA = {"Fa 1 hora", "Hace 1 hora", "1 hour ago"
100  };
101  /** Label del formulari resum: Fa {X} hores */
102  public static final String[] TEXT_FAXHORES = {"Fa {X} hores", "Hace {X} horas", "{X}
103  hours ago"};
104  /** Label del formulari resum: Fa 1 dia */
105  public static final String[] TEXT_FA1DIA = {"Fa 1 dia", "Hace 1 día", "1 day ago"};
106  /** Label del formulari resum: Fa {X} dies */
107  public static final String[] TEXT_FAXDIES = {"Fa {X} dies", "Hace {X} días", "{X} days
108  ago"};
109
110  /** Label del formulari configuració: Codi PPStop */
111  public static final String[] LABEL_CONFPPSTOP = {"Codi PPStop:", "Código PPStop:"
112  , "PPStop Code:"};
113  /** Label del formulari configuració: Tipus Alarma */
114  public static final String[] LABEL_CONFIPUSALARMA = {"Tipus Alarma:", "Tipo Alarma:"
115  , "Alarm Style:"};
116  /** Label del formulari configuració: Idioma */
117  public static final String[] LABEL_CONFIDIOMA = {"Idioma:", "Idioma:", "Language:"};
118  /** Label del formulari configuració: Temps Alarma */
119  public static final String[] LABEL_CONFTEMPALARMA = {"Temps Alarma:", "Tiempo Alarma
120  :", "Alarm Time:"};
121  /** Label del formulari configuració: Volum Alarma */
122  public static final String[] LABEL_CONFVOLALARMA = {"Volum Alarma:", "Volumen Alarma:
123  ", "Alarm Volume:"};
124  /** Label del formulari configuració: Sensibilitat */
125  public static final String[] LABEL_CONFSENSIBILITAT = {"Sensibilitat:", "Sensibilidad
126  :", "Sensitivity:"};
127  /** Label del formulari configuració: Català */
128  public static final String[] LABEL_CA = {"Català", "Catalán", "Catalan"};
129  /** Label del formulari configuració: Castellà */

```

```

114 public static final String[] LABEL_ES = {"Castellà", "Castellano", "Spanish"};
115 /** Label del formulari configuració: Anglès */
116 public static final String[] LABEL_EN = {"Anglès", "Inglés", "English"};
117 /** Label del formulari configuració: Veu Humana */
118 public static final String[] LABEL_VEUHUMANA = {"Veu humana", "Voz humana", "Human
voice"};
119 /** Label del formulari configuració: Ring */
120 public static final String[] LABEL_RING = {"Ring", "Ring", "Ring"};
121 /** Label del formulari configuració: Silenci */
122 public static final String[] LABEL_SILENCI = {"Silenci", "Silencio", "Mute"};
123 /** Label del formulari configuració: 10 seg */
124 public static final String[] LABEL_T10S = {"10 seg", "10 seg", "10 sec"};
125 /** Label del formulari configuració: 20 seg */
126 public static final String[] LABEL_T20S = {"20 seg", "20 seg", "20 sec"};
127 /** Label del formulari configuració: 30 seg */
128 public static final String[] LABEL_T30S = {"30 seg", "30 seg", "30 sec"};
129 /** Label del formulari configuració: Indefinit */
130 public static final String[] LABEL_TINDEFINIT = {"Indefinit", "Indefinido", "Infinite"
};
131 /** Label del formulari configuració: 20% */
132 public static final String[] LABEL_VOL20 = {"20%", "20%", "20%"};
133 /** Label del formulari configuració: 40% */
134 public static final String[] LABEL_VOL40 = {"40%", "40%", "40%"};
135 /** Label del formulari configuració: 60% */
136 public static final String[] LABEL_VOL60 = {"60%", "60%", "60%"};
137 /** Label del formulari configuració: 80% */
138 public static final String[] LABEL_VOL80 = {"80%", "80%", "80%"};
139 /** Label del formulari configuració: 100% */
140 public static final String[] LABEL_VOL100 = {"100%", "100%", "100%"};
141
142 /** Text de confirmació per sortir de l'aplicació */
143 public static final String[] TEXT_SORTIR = {"Sortir de l'aplicació?", "¿Salir de la
aplicación?", "Exit application?"};
144 /** Text d'informació de terminal no associat a cap servei PPStop */
145 public static final String[] TEXT_NOASSOCIAT = {"No estàs associat a cap servei
PPStop!", "¡No estás asociado a ningún servicio PPStop!", "You are not matched to
any PPStop service!"};
146 /** Text de confirmació per a búsqueda de terminals */
147 public static final String[] TEXT_CONFIRMACIOBUSCARTERMINALS = {"En la pròxima
connexió s'iniciarà una nova búsqueda i s'esborraran els terminals associats al
dispositiu PPStop. Continuar?", "En la próxima conexión se iniciará una nueva
búsqueda y se borrarán los terminales asociados al dispositivo PPStop. Continuar
?", "In next connexion, a new search will begin and PPStop matched devices will
be deleted. Continue?"};
148 /** Text de confirmació per a canvi de sensibilitat */
149 public static final String[] TEXT_CONFIRMACIOMODIFICARSENSIBILITAT = {"Selecciona la
nova sensibilitat. En la pròxima connexió es procedirà a canviar la
sensibilitat del dispositiu PPStop. Continuar?", "Selecciona la nueva
sensibilidad. En la próxima conexión se procederá a cambiar la sensibilidad del
dispositivo PPStop. ¿Continuar?", "Select the desired sensibility. The new
sensibility will be applied to PPStop device in next conection. Continue?"};
150 /** Text de confirmació per a rotació de erminals */
151 public static final String[] TEXT_CONFIRMACIOROTARTERMINALS = {"En la pròxima
connexió s'efectuarà una rotació cíclica de l'ordre dels terminals. Continuar?",
"En la próxima conexión se efectuará una rotación cíclica del orden de los
terminales. ¿Continuar?", "Devices order will be changed in next connexion.
Continue?"};
152 /** Text de confirmació de cancel·lar acció (resetejar flag) */
153 public static final String[] TEXT_CANCELARACCIO = {"Aquesta acció ja està activa.
Vols cancel·lar-la?", "Esta acción ya está activa. ¿Quieres cancelarla?", "This
option is already active. Do you want to cancel it?"};
154 /** Text d'informació d'historial buit */
155 public static final String[] TEXT_NOALARMES = {"L'historial està buit!", "¡El
historial está vacío!", "The history is empty!"};
156 /** Text de confirmació per a esborrar l'historial */
157 public static final String[] TEXT_RESETHISTORIAL = {"S'esborrarà l'historial d'
alarmes. N'estàs segur?", "Se borrará todo el historial de alarmas. ¿Estás seguro
?", "Alarm history will be deleted. Are you sure?"};
158 /** Text d'informació de dispositiu associat correctament */
159 public static final String[] TEXT_DISPASSOCIAT = {"Dispositiu associat correctament!

```

```

160     ", "Dispositivo asociado correctamente!", "Device succesfully matched!");
161 /** Text d'informació de sensibilitat modificada correctament */
162 public static final String[] TEXT_SENSCANVIADA = {"Sensibilitat modificada
163 correctament!", "Sensibilidad modificada correctamente!", "Sensitivity changed
164 successfully!"};
165 /** Text d'informació de dispositiu associat correctament */
166 public static final String[] TEXT_TERMINALROTAT = {"La rotació de terminals s'ha
167 efectuat correctament!", "La rotación de terminales se ha realizado
168 correctamente!", "Device rotation has been succesfull!"};
169 /** Text d'informació de transferir historial */
170 public static final String[] TEXT_TRANSFERIRHISTORIAL = {"El codi que has d'
171 introduir a l'Aplicació de Seguiment PPStop per a poder transferir l'historial d'
172 alarmes és: ", "El cdig o que tienes que introducir en la Aplicación de
173 Seguimiento PPStop para poder transferir el historial de alarmas es: ", "The code
174 you have to put in the PPStop Monitoring Application in order to transfer the
175 Alarm History is: "};
176 /** Text d'informació de connectivitat bluetooth desactivada */
177 public static final String[] TEXT_NOBTACTIU = {"No tens activa la connectivitat
178 Bluetooth del teu terminal!", "No tienes activada la conectividad Bluetooth de
179 tu terminal!", "Your device bluetooth conectivity is innactive!"};
180 /** Text d'informació d'historial d'alarmes transferit correctament */
181 public static final String[] TEXT_HISTTRANSFERIT = {"Historial d'alarmes transferit
182 correctament!", "Historial de alarmas transferido correctamente!", "Alarm history
183 succesfully transmitted!"};
184 /** Text d'informació de búsqueda en progrés */
185 public static final String[] TEXT_BUSQUEDAPROGRES = {"Esperi, búsqueda de terminals
186 en progrés...", "Espere, realizando la búsqueda de terminales...", "Wait, device
187 search in progress..."};
188 }

```

### Codi C.7: (AppControl) BD.java

```

1 package Dades;
2
3 import Constants.CONST;
4 import java.io.IOException;
5
6 /**
7  * Classe que actua de gestor de Base de Dades de l'aplicació
8  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
9  * @version 3.1 (Maig de 2010)
10  */
11 public class BD {
12
13     /** Objecte de Record Management Store */
14     private RMS RMS = new RMS();
15
16     /** Crea una instància de BD */
17     public BD(){}
18
19     /**
20      * Obté la configuració de l'aplicació emmagatzemada a la BD
21      * @return Configuració recuperada
22      */
23     public Config getConfig(){
24         Config C = new Config();
25         try {
26             byte[] b = RMS.getRS(CONST.RS_CONFIG);
27             if (b!=null)
28                 C.fromByteArray(b);
29             } catch (IOException ex) {}
30         return C;
31     }
32
33     /**
34      * Emmagatzema la configuració de l'aplicació a la BD
35      * @param C Configuració a emmagatzemar
36      */
37     public void setConfig(Config C){

```

```

38         try {
39             RMS.setRS(CONST.RS_CONFIG, C.toByteStream());
40         } catch (IOException ex) {}
41     }
42
43     /**
44     * Esborra la configuració emmagatzemada a la BD
45     */
46     public void delConfig(){
47         RMS.delRS(CONST.RS_CONFIG);
48     }
49
50     /**
51     * Obté l'historial d'alarmes emmagatzemat a la BD
52     * @return Historial d'Alarmes recuperat
53     */
54     public Historial getHistorial(){
55         Historial H = new Historial();
56         try {
57             byte[] b = RMS.getRS(CONST.RS_HISTORIAL);
58             if (b!=null)
59                 H.fromByteStream(b);
60         } catch (IOException ex) {}
61         return H;
62     }
63
64     /**
65     * Emmagatzema l'historial d'alarmes a la BD
66     * @param H      Historial d'Alarmes a emmagatzemar
67     */
68     public void setHistorial(Historial H){
69         try {
70             RMS.setRS(CONST.RS_HISTORIAL, H.toByteStream());
71         } catch (IOException ex) {}
72     }
73
74     /**
75     * Esborra l'historial d'alarmes emmagatzemat a la BD
76     */
77     public void delHistorial(){
78         RMS.delRS(CONST.RS_HISTORIAL);
79     }
80
81 }

```

Codi C.8: (AppControl) Config.java

```

1 package Dades;
2
3 import Constants.CONST;
4 import java.io.ByteArrayInputStream;
5 import java.io.ByteArrayOutputStream;
6 import java.io.DataInputStream;
7 import java.io.DataOutputStream;
8 import java.io.IOException;
9
10 /**
11  * Classe representativa de la configuració de l'aplicació
12  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
13  * @version 3.1 (Maig de 2010)
14  */
15 public class Config {
16
17     /** Codi PPStop */
18     private String codi;
19     /** Identificador de l'estat del servei PPStop */
20     private int estat;
21     /** Estat de la bateria del dispositiu PPStop (%) */
22     private int bateria;

```

```

23  /** Sensibilitat del sensor d'humitat del dispositiu PPStop (1-5) */
24  private int sensibilitat;
25  /** Ordre del terminal en el sistema */
26  private int ordre;
27  /** Idioma de la interfície de l'aplicació */
28  private int idioma;
29  /** Identificador del tpus d'alarma de l'aplicació */
30  private int tipusalarma;
31  /** Temps màxim d'alarma (ms) */
32  private int maxtempsalarma;
33  /** Nivell de volum de l'alarma (%) */
34  private int volumalarma;
35
36  /**
37   * Crea una instància de Config, amb la configuració per defecte
38   */
39  public Config(){
40      codi = CONST.EMPTY;
41      estat = CONST.ESTAT_NOASSOCIAT;
42      bateria = CONST.NODEF;
43      sensibilitat = CONST.NODEF;
44      ordre = CONST.NODEF;
45      idioma = CONST.IDIOMA_CA;
46      tipusalarma = CONST.ALARMA_VEUHUMANA;
47      maxtempsalarma = CONST.ALARMA_TIOS;
48      volumalarma = CONST.ALARMA_VOL100;
49  }
50
51  /**
52   * Obté el codi PPStop
53   * @return Codi PPStop
54   */
55  public String getCodi(){
56      return codi.toUpperCase();
57  }
58
59  /**
60   * Guarda el codi PPStop
61   * @param valor      Codi PPStop
62   */
63  public void setCodi(String valor){
64      this.codi = valor;
65  }
66
67  /**
68   * Obté l'estat del servei PPStop
69   * @return Identificador de l'estat del servei
70   */
71  public int getEstat(){
72      return estat;
73  }
74
75  /**
76   * Guarda l'estat del servei PPStop
77   * @param valor      Identificador de l'estat del servei
78   */
79  public void setEstat(int valor){
80      estat = valor;
81  }
82
83  /**
84   * Obté l'estat de la bateria del dispositiu PPStop (%)
85   * @return Estat de la bateria (%)
86   */
87  public int getBateria(){
88      return bateria;
89  }
90
91  /**
92   * Guarda l'estat de la bateria del dispositiu PPStop (%)

```



```

93     * @param valor      Estat de la bateria (%)
94     */
95     public void setBateria(int valor){
96         bateria = valor;
97     }
98
99     /**
100     * Obte la sensibilitat del sensor d'humitat del dispositiu PPStop
101     * @return Sensibilitat del sensor (1-5)
102     */
103     public int getSensibilitat(){
104         return sensibilitat;
105     }
106
107     /**
108     * Guarda la sensibilitat del sensor d'humitat del dispositiu PPStop
109     * @param valor      Sensibilitat del sensor (0:baix/1:alt)
110     */
111     public void setSensibilitat(int valor){
112         sensibilitat = valor;
113     }
114
115     /**
116     * Obte l'ordre del terminal en el sistema
117     * @return Ordre del terminal (cardinal)
118     */
119     public int getOrdre(){
120         return ordre;
121     }
122
123     /**
124     * Guarda l'ordre del terminal en el sistema
125     * @param valor      Ordre del terminal (cardinal)
126     */
127     public void setOrdre(int valor){
128         ordre = valor;
129     }
130
131     /**
132     * Obté l'idioma de la interfície de l'aplicació
133     * @return Identificador de l'idioma
134     */
135     public int getIdioma(){
136         return idioma;
137     }
138
139     /**
140     * Guardar l'idioma de la interfície de l'aplicació
141     * @param idioma      Identificador de l'idioma
142     */
143     public void setIdioma(int idioma){
144         this.idioma = idioma;
145     }
146
147     /**
148     * Obté el tipus d'alarma de l'aplicació
149     * @return Identificador del tipus d'alarma
150     */
151     public int getTipusAlarma(){
152         return tipusalarma;
153     }
154
155     /**
156     * Guarda el tipus d'alarma de l'aplicació
157     * @param valor      Identificador del tipus d'alarma
158     */
159     public void setTipusAlarma(int valor){
160         tipusalarma = valor;
161     }
162

```



```

163  /**
164   * Obté el temps màxim d'alarma
165   * @return Temps màxim d'alarma (ms)
166   */
167  public int getMaxTempsAlarma(){
168      return maxtempsalarma;
169  }
170
171  /**
172   * Guarda el temps màxim d'alarma
173   * @param valor      Temps màxim d'alarma (ms)
174   */
175  public void setMaxTempsAlarma(int valor){
176      maxtempsalarma = valor;
177  }
178
179  /**
180   * Obté el volum de l'alarma
181   * @return Nivell de volum de l'alarma (%)
182   */
183  public int getVolumAlarma(){
184      return volumalarma;
185  }
186
187  /**
188   * Guarda el volum de l'alarma
189   * @param valor      Nivell de volum de l'alarma (%)
190   */
191  public void setVolumAlarma(int valor){
192      volumalarma = valor;
193  }
194
195  /**
196   * Serialitza l'objecte
197   * @return Stream de bytes (Objecte serialitzat)
198   */
199  public byte[] toByteStream() throws IOException{
200      ByteArrayOutputStream baos = new ByteArrayOutputStream();
201      DataOutputStream dos = new DataOutputStream(baos);
202      dos.writeUTF(codi);
203      dos.writeUTF(String.valueOf(estat));
204      dos.writeUTF(String.valueOf(bateria));
205      dos.writeUTF(String.valueOf(sensibilitat));
206      dos.writeUTF(String.valueOf(ordre));
207      dos.writeUTF(String.valueOf(idioma));
208      dos.writeUTF(String.valueOf(tipusalarma));
209      dos.writeUTF(String.valueOf(maxtempsalarma));
210      dos.writeUTF(String.valueOf(volumalarma));
211      dos.flush();
212      return baos.toByteArray();
213  }
214
215  /**
216   * Des-Serialitza l'objecte
217   * @param dades      Stream de bytes (Objecte des-serialitzat)
218   */
219  public void fromByteStream(byte[] dades) throws IOException{
220      ByteArrayInputStream bais = new ByteArrayInputStream(dades);
221      DataInputStream dis = new DataInputStream(bais);
222      codi = dis.readUTF();
223      estat = Integer.parseInt(dis.readUTF());
224      bateria = Integer.parseInt(dis.readUTF());
225      sensibilitat = Integer.parseInt(dis.readUTF());
226      ordre = Integer.parseInt(dis.readUTF());
227      idioma = Integer.parseInt(dis.readUTF());
228      tipusalarma = Integer.parseInt(dis.readUTF());
229      maxtempsalarma = Integer.parseInt(dis.readUTF());
230      volumalarma = Integer.parseInt(dis.readUTF());
231  }
232

```

## Codi C.9: (AppControl) Historial.java

```

1 package Dades;
2
3 import java.io.ByteArrayInputStream;
4 import java.io.ByteArrayOutputStream;
5 import java.io.DataInputStream;
6 import java.io.DataOutputStream;
7 import java.io.IOException;
8 import java.util.Enumeration;
9 import java.util.TimeZone;
10 import java.util.Vector;
11
12 /**
13  * Classe representativa de l'Historial d'Alarmes
14  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
15  * @version 3.1 (Maig de 2010)
16  */
17 public class Historial {
18
19     /** Vector de timestamps */
20     private Vector V;
21
22     /**
23      * Crea una instància de Historial
24      */
25     public Historial(){
26         V = new Vector();
27     }
28
29     /**
30      * Obté el vector amb l'Historial d'Alarmes
31      * @return Vector de timestamps
32      */
33     public Vector getHistorial(){
34         return V;
35     }
36
37     /**
38      * Guarda el vector amb l'Historial d'Alarmes
39      * @param valor Vector de timestamps
40      */
41     public void setHistorial(Vector valor){
42         V = valor;
43     }
44
45     /**
46      * Afegeix una nova alarma a l'Historial
47      * @param valor Timestamp de l'alarma a afegir
48      */
49     public void setNovaAlarma(String valor){
50         V.addElement(valor);
51     }
52
53     /**
54      * Reseteja l'Historial d'Alarmes
55      */
56     public void delHistorial(){
57         V.removeAllElements();
58     }
59
60     /**
61      * Serialitza l'objecte
62      * @return Stream de bytes (Objecte serialitzat)
63      */
64     public byte[] toByteStream() throws IOException{
65         ByteArrayOutputStream baos = new ByteArrayOutputStream();

```

```

66      DataOutputStream dos = new DataOutputStream(baos);
67      dos.writeInt(V.size());
68      dos.writeUTF(TimeZone.getDefault().getID());
69      Enumeration vEnum = V.elements();
70      while(vEnum.hasMoreElements())
71          dos.writeUTF((String)vEnum.nextElement());
72      dos.flush();
73      return baos.toByteArray();
74  }
75
76  /**
77   * Des-Serialitza l'objecte
78   * @param dades      Stream de bytes (Objecte des-serialitzat)
79   */
80  public void fromByteStream(byte[] dades) throws IOException{
81      ByteArrayInputStream bais = new ByteArrayInputStream(dades);
82      DataInputStream dis = new DataInputStream(bais);
83      int n = dis.readInt();
84      dis.readUTF(); //TZid
85      V = new Vector();
86      while ((n--)>0)
87          V.addElement(dis.readUTF());
88  }
89
90 }

```

### Codi C.10: (AppControl) Model.java

```

1 package Dades;
2
3 import Audio.Alarma;
4 import Constants.CONST;
5 import Utils.Utils;
6 import Vista.Vista;
7 import java.util.Enumeration;
8 import java.util.Vector;
9
10 /**
11  * Classe que conté totes les dades de l'aplicació. Actua com a Model dins del patró MVC
12  * (Model-View-Controller)
13  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
14  * @version 3.1 (Maig de 2010)
15  */
16 public class Model {
17     /** Referència a la Vista */
18     private Vista V;
19     /** Referència a l'Alarma Sonora */
20     private Alarma A;
21     /** Objecte que conté la Configuració */
22     private Config C;
23     /** Objecte que conté l'Historial d'alarmes */
24     private Historial H;
25     /** Objecte Gestor de la BD */
26     private BD BD = new BD();
27     /** Objecte amb funcions d'Utilitat per al tractament de dades */
28     private Utils U = new Utils();
29     /**Adreça bluetooth del terminal */
30     private String macMobil;
31     /** Flag de servei */
32     private int flag;
33     /** Identificador de l'estat de la connectivitat bluetooth del terminal */
34     private int estatBT;
35     /** Nova sensibilitat del sensor d'humitat del dispositiu PPStop (1-5) a TX */
36     private int sensibilitatTX;
37     /** Indica si la Vista ha de ser refrescada */
38     private boolean refresh;
39
40     /**

```

```

41     * Crea una instància de Model, amb els valors per defecte
42     */
43     public Model(){
44         resetValors();
45         C = BD.getConfig();
46         H = BD.getHistorial();
47     }
48
49     /**
50     * Associa la vista al model (patró MVC)
51     * @param V      Referència a la Vista
52     */
53     public void setVista(Vista V){
54         this.V = V;
55     }
56
57     /**
58     * Associa l'alarma al model (patró MVC)
59     * @param A      Referència a l'Alarma Sonora
60     */
61     public void setAlarma(Alarma A){
62         this.A = A;
63     }
64
65     /**
66     * Obté l'adreça bluetooth del terminal mòbil
67     * @return Adreça bluetooth
68     */
69     public String getMacMovil() {
70         return macMobil;
71     }
72
73     /**
74     * Guarda l'adreça bluetooth del terminal mòbil
75     * @param valor    Adreça bluetooth
76     */
77     public void setMacMovil(String valor) {
78         macMobil = valor.toUpperCase();
79     }
80
81     /**
82     * Obté el flag de servei
83     * @return Identificador del flag de servei
84     */
85     public int getFlag(){
86         return flag;
87     }
88
89     /**
90     * Estableix el flag de servei
91     * @param valor    Identificador del flag de servei
92     */
93     public void setFlag(int valor){
94         flag = valor;
95     }
96
97     /**
98     * Obté l'estat de la connectivitat Bluetooth del terminal
99     * @return Identificador de l'estat de la connectivitat
100    */
101    public int getEstatBT(){
102        return estatBT;
103    }
104
105    /**
106    * Estableix l'estat de la connectivitat Bluetooth del terminal
107    * @param valor    Identificador de l'estat de la connectivitat
108    */
109    public void setEstatBT(int valor){
110        estatBT = valor;

```

```

111         V.update();
112     }
113
114     /**
115     * Obte l'estat del servei PPStop
116     * @return Identificador de l'estat del servei
117     */
118     public int getEstat(){
119         return C.getEstat();
120     }
121
122     /**
123     * Estableix l'estat del servei PPStop
124     * @param valor      Identificador de l'estat del servei
125     */
126     public void setEstat(int valor){
127         C.setEstat(valor);
128         BD.setConfig(C);
129         V.update();
130     }
131
132     /**
133     * Obté l'estat de la bateria del dispositiu PPStop (%)
134     * @return Estat de la bateria (%)
135     */
136     public int getBateria(){
137         return C.getBateria();
138     }
139
140     /**
141     * Guarda l'estat de la bateria del dispositiu PPStop (%)
142     * @param valor      Estat de la bateria (%)
143     */
144     public void setBateria(int valor){
145         C.setBateria(valor);
146         BD.setConfig(C);
147         V.update();
148     }
149
150     /**
151     * Obté la sensibilitat del sensor d'humitat del dispositiu PPStop
152     * @return Sensibilitat del sensor (0:baix/1:alt)
153     */
154     public int getSensibilitat(){
155         return C.getSensibilitat();
156     }
157
158     /**
159     * Guarda la sensibilitat del sensor d'humitat del dispositiu PPStop
160     * @param valor      Sensibilitat del sensor (0:baix/1:alt)
161     */
162     public void setSensibilitat(int valor){
163         C.setSensibilitat(valor);
164         BD.setConfig(C);
165         V.update();
166     }
167
168     /**
169     * Obté la nova sensibilitat del sensor d'humitat a TX
170     * @return Sensibilitat del sensor a TX (0:baix/1:alt)
171     */
172     public int getSensibilitatTX(){
173         return sensibilitatTX;
174     }
175
176     /**
177     * Estableix la nova sensibilitat del sensor d'humitat a TX
178     * @param valor      Sensibilitat del sensor a TX (0:baix/1:alt)
179     */
180     public void setSensibilitatTX(int valor){

```

```

181         sensibilitatTX = valor;
182     }
183
184     /**
185     * Obté l'ordre del terminal en el sistema
186     * @return Ordre del terminal (cardinal)
187     */
188     public int getOrdre(){
189         return C.getOrdre();
190     }
191
192     /**
193     * Guarda l'ordre del terminal en el sistema
194     * @param valor      Ordre del terminal (cardinal)
195     */
196     public void setOrdre(int valor){
197         C.setOrdre(valor);
198         BD.setConfig(C);
199         V.update();
200     }
201
202     /**
203     * Guarda els paràmetres generals del sistema: estat de la bateria, nivell de
204     * sensibilitat, ordre del terminal en el sistema
205     * @param valor      Estat de la bateria (%)
206     * @param valor      Sensibilitat del sensor (0:baix/1:alt)
207     * @param valor      Ordre del terminal (cardinal)
208     */
209     public void setParametres(int bateria, int sensibilitat, int ordre){
210         C.setBateria(bateria);
211         C.setSensibilitat(sensibilitat);
212         C.setOrdre(ordre);
213         BD.setConfig(C);
214         V.update();
215     }
216
217     /**
218     * Obté el codi PPStop
219     * @return Codi PPStop
220     */
221     public String getCodi(){
222         return C.getCodi();
223     }
224
225     /**
226     * Estableix el codi PPStop
227     * @param valor      Codi PPStop
228     */
229     public void setCodi(String valor){
230         C.setCodi(valor.toUpperCase());
231         BD.setConfig(C);
232         V.update();
233     }
234
235     /**
236     * Obté l'idioma de la interfície de l'aplicació
237     * @return Identificador de l'idioma
238     */
239     public int getIdioma(){
240         return C.getIdioma();
241     }
242
243     /**
244     * Estableix l'idioma de la interfície de l'aplicació
245     * @param valor      Identificador de l'idioma
246     */
247     public void setIdioma(int valor){
248         C.setIdioma(valor);
249         BD.setConfig(C);
250         V.update();

```

```

250     }
251
252     /**
253      * Obté el tipus d'alarma de l'aplicació
254      * @return Identificador del tipus d'alarma
255      */
256     public int getTipusAlarma(){
257         return C.getTipusAlarma();
258     }
259
260     /**
261      * Estableix el tipus d'alarma de l'aplicació
262      * @param valor      Identificador del tipus d'alarma
263      */
264     public void setTipusAlarma(int valor){
265         C.setTipusAlarma(valor);
266         BD.setConfig(C);
267         A.update();
268     }
269
270     /**
271      * Obté el temps màxim d'alarma
272      * @return Temps màxim d'alarma (ms)
273      */
274     public int getMaxTempsAlarma(){
275         return C.getMaxTempsAlarma();
276     }
277
278     /**
279      * Estableix el temps màxim d'alarma (ms)
280      * @param valor      Temps màxim d'alarma (ms)
281      */
282     public void setMaxTempsAlarma(int valor){
283         C.setMaxTempsAlarma(valor);
284         BD.setConfig(C);
285     }
286
287     /**
288      * Obté el volum de l'alarma
289      * @return Nivell de volum de l'alarma (%)
290      */
291     public int getVolumAlarma(){
292         return C.getVolumAlarma();
293     }
294
295     /**
296      * Estableix el volum de l'alarma
297      * @param valor      Nivell de volum de l'alarma (%)
298      */
299     public void setVolumAlarma(int valor){
300         C.setVolumAlarma(valor);
301         BD.setConfig(C);
302         A.setVolum(valor);
303     }
304
305     /**
306      * Obté si la vista ha de ser refrescada
307      * @return Valor booleà
308      */
309     public boolean getRefresh(){
310         return refresh;
311     }
312
313     /**
314      * Estableix si la vista ha de ser refrescada
315      * @param valor      Valor booleà
316      */
317     public void setRefresh(boolean valor){
318         refresh = valor;
319     }

```

```

320
321 /**
322  * Reseteja els valors del model als valors per defecte
323  */
324 private void resetValors() {
325     flag = CONST.FLAG_ALARMA;
326     estatBT = CONST.ESTAT_DESACTIVAT;
327     sensibilitatTX = CONST.NODEF;
328     refresh = true;
329 }
330
331 /**
332  * Obté l'objecte Historial d'Alarmes
333  * @return Objecte Historial
334  */
335 public Historial getHistorial(){
336     return H;
337 }
338
339 /**
340  * Obté l'Historial d'Alarmes en format TXT
341  * @return Vector d'alarmes en format TXT
342  */
343 public Vector getHistorialTXT(){
344     Vector vec = new Vector();
345     Vector v = H.getHistorial();
346     Enumeration vecEnum = v.elements();
347     while (vecEnum.hasMoreElements())
348         vec.addElement(U.obtenirData(Long.parseLong(vecEnum.nextElement().toString())
349             ),CONST.DATA_LLARGA));
350     U.invertirVector(vec);
351     return vec;
352 }
353
354 /**
355  * Obté la darrera alarma de l'Historial d'Alarmes en format TXT
356  * @return Darrera alarma en format TXT
357  */
358 public String getDarreraAlarma(){
359     Vector vec = H.getHistorial();
360     String s = null;
361     if (!vec.isEmpty())
362         s = U.obtenirData(Long.parseLong(vec.elementAt(vec.size()-1).toString()),
363             CONST.DATA_CURTA);
364     return s;
365 }
366
367 /**
368  * Obté la diferència de temps amb la darrera alarma de l'Historial d'Alarmes en
369     format TXT
370  * @return Diferència de temps en format TXT
371  */
372 public String getDarreraAlarmaDif(){
373     Vector vec = H.getHistorial();
374     String s = null;
375     if (!vec.isEmpty())
376         s = U.obtenirDiferenciaData(Long.parseLong((String)vec.lastElement()),C.
377             getIdioma());
378     return s;
379 }
380
381 /**
382  * Registra una nova alarma a l'Historial d'Alarmes
383  */
384 public void registrarAlarma(){
385     H.setNovaAlarma(String.valueOf(U.obtenirTimestamp()));
386     BD.setHistorial(H);
387     V.update();
388 }

```



```

386     /**
387     * Reseteja l'Historial d'Alarmes
388     */
389     public void resetHistorial(){
390         H.delHistorial();
391         BD.delHistorial();
392         V.update();
393     }
394 }
395 }

```

### Codi C.11: (AppControl) RMS.java

```

1 package Dades;
2
3 import javax.microedition.rms.*;
4
5 /**
6  * Classe que gestiona la RMS (Record Manager Store)
7  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
8  * @version 3.1 (Maig de 2010)
9  */
10 public class RMS {
11
12     /** Objecte de Record Store */
13     private RecordStore RS = null;
14
15     /** Crea una instància de RMS */
16     public RMS(){
17
18     /**
19      * Obre un Record Store
20      * @param RS Identificador del Record Store a obrir
21      */
22     private void openRS(String RS){
23         try {
24             this.RS = RecordStore.openRecordStore(RS, true);
25         } catch (Exception e){}
26     }
27
28     /**
29      * Tanca un Record Store
30      */
31     private void closeRS(){
32         try {
33             RS.closeRecordStore();
34         } catch (Exception e){}
35     }
36
37     /**
38      * Escriu un Record en el Record Store obert
39      * @param b Flux de dades a escriure
40      */
41     private void writeRecord(byte[] b){
42         try {
43             if (RS.getNumRecords()==0){
44                 RS.addRecord(b, 0, b.length);
45             } else {
46                 RS.setRecord(1, b, 0, b.length);
47             }
48         } catch (Exception e){}
49     }
50
51     /**
52      * Llegeix un Record del Record Store obert
53      * @return Flux de dades llegit
54      */
55     private byte[] readRecord(){
56         /* Llegir Record */

```

```

57         byte[] b = null;
58         try {
59             if (RS.getNumRecords()>0)
60                 b = RS.getRecord(1);
61         } catch (Exception e){};
62         return b;
63     }
64
65     /**
66      * Llegeix un flux de bytes del Record Store
67      * @param RS    Record Store sobre el que aplicar la lectura
68      * @return      Flux de bytes llegits
69      */
70     public byte[] getRS(String RS){
71         openRS(RS);
72         byte[] b = readRecord();
73         closeRS();
74         return b;
75     }
76
77     /**
78      * Guarda un flux de bytes al Record Store
79      * @param RS    Record Store sobre el que aplicar la escriptura
80      * @param b      Flux de bytes a escriure
81      */
82     public void setRS(String RS, byte[] b){
83         delRS(RS);
84         openRS(RS);
85         writeRecord(b);
86         closeRS();
87     }
88
89     /**
90      * Elimina un Record Store
91      * @param RS    Record Store a eliminar
92      */
93     public void delRS(String RS){
94         try {
95             if (RecordStore.listRecordStores()!=null)
96                 RecordStore.deleteRecordStore(RS);
97         } catch (Exception e){}
98     }
99
100 }

```

### Codi C.12: (AppControl) PPStop.java

```

1 package Main;
2
3 import Constants.CONST;
4 import Dades.Model;
5 import Audio.Audio;
6 import Audio.Alarma;
7 import Bluetooth.Servei;
8 import Constants.LANG;
9 import Vista.Vista;
10 import com.sun.lwuit.Command;
11 import java.util.Timer;
12 import java.util.TimerTask;
13 import javax.microedition.lcdui.Display;
14 import javax.microedition.midlet.*;
15
16
17 /**
18  * Classe principal de l'aplicació PPStop
19  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
20  * @version 3.1 (Maig de 2010)
21  */
22 public class PPStop extends MIDlet {

```

```

23
24 /** Model de dades */
25 private Model M;
26 /** Vista del MIDlet */
27 private Vista V;
28 /** Alarma Sonora */
29 private Alarma A;
30 /** Servei PPStop */
31 private Servei S;
32 /** Objecte per a timeouts */
33 private Timer T;
34
35 /**
36  * Crea una instància de PPStop
37  */
38 public PPStop(){}
39
40 /**
41  * Crea tots els elements del patró Model-View-Controller (MVC) i inicialitza el
42  * MIDlet
43  */
44 public void startApp() {
45     M = new Model();
46     V = new Vista(this,M);
47     M.setVista(V);
48     A = new Alarma(M);
49     M.setAlarma(A);
50     S = new Servei(this,M);
51     initMIDlet();
52 }
53
54 /**
55  * Activa l'alarma PPStop
56  */
57 public void StartAlarma(){
58     Display.getDisplay(this).flashBacklight(60000);
59     V.mostrarFormulari(CONST.FORM_ALARMA);
60     if (M.getTipusAlarma() != CONST.ALARMA_SILENCI)
61         A.playAlarma();
62     M.registrarAlarma();
63     T = new Timer();
64     if (M.getMaxTempsAlarma() > 0)
65         T.schedule(new endTimerAlarma(),M.getMaxTempsAlarma());
66 }
67
68 /**
69  * Atura l'alarma PPStop
70  */
71 public void StopAlarma(){
72     T.cancel();
73     if (M.getTipusAlarma() != CONST.ALARMA_SILENCI)
74         A.stopAlarma();
75     Display.getDisplay(this).flashBacklight(0);
76     V.mostrarFormulari(CONST.FORM_ANTERIOR);
77 }
78
79 /**
80  * Final del Timer d'Alarma PPStop: Atura l'alarma
81  */
82 private class endTimerAlarma extends TimerTask {
83     public void run(){
84         StopAlarma();
85     }
86 }
87
88 /**
89  * Reprodueix instantàniament un arxiu d'àudio
90  * @param arxiu Arxiu d'àudio a reproduir
91  * @param tipus Tipus d'arxiu d'àudio a reproduir
92  */

```

```

92     public void PlayAudio(String arxiu, String tipus){
93         new Audio(arxiu,tipus,100,1,true).start();
94     }
95
96     /**
97     * Mostra un dialog informatiu durant un cert timeout
98     * @param tipus      Tipus de Dialog a mostrar
99     * @param text       Contingut del Dialog
100    * @param timeout    Timeout del Dialog (ms)
101    * @param height     Alçada del Dialog (px)
102    */
103    public void mostrarDialogInformatiu(final int tipus, final String text, final long
        timeout, final int height){
104        Display.getDisplay(this).flashBacklight(1);
105        new Thread(new Runnable() {
106            public void run() {
107                Command cmd = null;
108                if (timeout==0)
109                    cmd = new Command(LANG.MENU_OK[M.getIdioma()],CONST.COMMAND_RES);
110                V.mostrarDialog(tipus, text, cmd, null, timeout, height);
111            }
112        }).start();
113    }
114
115    /**
116    * Inicialitza el servei bluetooth PPStop
117    */
118    private void initMIDlet(){
119        S.start();
120    }
121
122    /**
123    * Notifica la sortida del MIDlet
124    */
125    public void exitMIDlet(){
126        notifyDestroyed();
127    }
128
129    /**
130    * Notifica la pausa del MIDlet
131    */
132    public void pauseApp() {
133        notifyPaused();
134    }
135
136    /**
137    * Sortir del MIDlet: destruir-lo
138    */
139    public void destroyApp(boolean unconditional) {
140        exitMIDlet();
141    }
142
143    /**
144    * Mètode auxiliar de LOG per a funcions de Debugging
145    * @param msg      Missatge a mostrar per pantalla
146    */
147    public void log(String msg) {
148        System.out.println(msg);
149    }
150
151 }

```

Codi C.13: (AppControl) Utils.java

```

1 package Utils;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import java.util.Calendar;

```

```

6 import java.util.Date;
7 import java.util.TimeZone;
8 import java.util.Vector;
9
10 /**
11  * Classe amb funcions útils d'àmbit general
12  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
13  * @version 3.1 (Maig de 2010)
14  */
15 public class Utils {
16
17     /** Crea una instància de Utils */
18     public Utils(){
19
20     /**
21      * Obte el timestamps actual
22      * @return Timestamp actual
23      */
24     public long obtenirTimestamp(){
25         return System.currentTimeMillis();
26     }
27
28     /**
29      * Obté la data en un cert format a partir d'un timestamp
30      * @param timestamp    Timestamp a convertir
31      * @param format        Format de la data a obtenir
32      * @return              Data formatada
33      */
34     public String obtenirData(long timestamp, int format){
35         Calendar cal = Calendar.getInstance(TimeZone.getDefault());
36         cal.setTime(new Date(timestamp));
37         String cal_YEAR = String.valueOf(cal.get(Calendar.YEAR));
38         String cal_MONTH = ((cal.get(Calendar.MONTH)+1)<10?"0":"") + String.valueOf(cal.
            get(Calendar.MONTH)+1);
39         String cal_DAY = (cal.get(Calendar.DAY_OF_MONTH)<10?"0":"") + String.valueOf(cal
            .get(Calendar.DAY_OF_MONTH));
40         String cal_HOUR = (cal.get(Calendar.HOUR_OF_DAY)<10?"0":"") + String.valueOf(cal
            .get(Calendar.HOUR_OF_DAY));
41         String cal_MIN = (cal.get(Calendar.MINUTE)<10?"0":"") + String.valueOf(cal.get(
            Calendar.MINUTE));
42         String data = null;
43         switch(format){
44             case CONST.DATA_CURTA:
45                 data = cal_DAY + "/" + cal_MONTH + "/" + cal_YEAR.substring(2,4) + " " +
                    cal_HOUR + ":" + cal_MIN;
46                 break;
47             case CONST.DATA_LLARGA:
48                 data = cal_DAY + "/" + cal_MONTH + "/" + cal_YEAR + " - " + cal_HOUR + "
                    : " + cal_MIN + "h";
49                 break;
50         }
51         return data;
52     }
53
54     /**
55      * Obté la diferència entre la data actual i una data passada en format TXT
56      * @param timestamp    Timestamp de la data passada
57      * @param idioma        Idioma desitjat
58      * @return              Diferència en format TXT
59      */
60     public String obtenirDiferenciaData(long timestamp, int idioma){
61         long dif = obtenirTimestamp()-timestamp;
62         if (dif < 60*1000){
63             return LANG.TEXT_FAMENYSUNMINUT[idioma];
64         } else if (dif < 120*1000){
65             return LANG.TEXT_FA1MINUT[idioma];
66         } else if (dif < 60*60*1000){
67             return substituirStr(LANG.TEXT_FAXMINUTS[idioma], "{X}", String.valueOf((int)
                Math.floor(dif/(60*1000))));
68         } else if (dif < 120*60*1000){

```

```

69         return LANG.TEXT_FA1HORA[idioma];
70     } else if (dif < 24*60*60*1000){
71         return substituirStr(LANG.TEXT_FAXHORES[idioma], "{X}", String.valueOf((int)
            Math.floor(dif/(3600*1000))));
72     } else if (dif < 48*60*60*1000){
73         return LANG.TEXT_FA1DIA[idioma];
74     } else {
75         return substituirStr(LANG.TEXT_FAXDIES[idioma], "{X}", String.valueOf((int)
            Math.floor(dif/(86400*1000))));
76     }
77 }
78
79 /**
80  * Inverteix els elements d'un vector
81  * @param vec    Vector a invertir
82  * @return       Vector invertit
83  */
84 public Vector invertirVector(Vector vec){
85     Object temp;
86     for(int i=0; i < vec.size()/2; i++){
87         temp = vec.elementAt(vec.size()-(i+1));
88         vec.setElementAt(vec.elementAt(i), vec.size()-(i+1));
89         vec.setElementAt(temp, i);
90     }
91     return vec;
92 }
93
94 /**
95  * Reemplaça strings d'una cadena de caràcters
96  * @param text    Text sobre el qual s'aplica la substitució
97  * @param comodi   Cadena de text a substituir
98  * @param substitut Cadena de text substituïda
99  * @return        Text amb la substitució aplicada
100 */
101 public static String substituirStr(String text, String comodi, String substitut) {
102     StringBuffer SB = new StringBuffer();
103     int startPos = 0;
104     int searchStringPos = text.indexOf(comodi);
105     int searchStringLength = comodi.length();
106     while(searchStringPos!=-1){
107         SB.append(text.substring(startPos, searchStringPos)).append(substitut);
108         startPos = searchStringPos + searchStringLength;
109         searchStringPos = text.indexOf(comodi, startPos);
110     }
111     return SB.append(text.substring(startPos, text.length())).toString();
112 }
113
114 }

```

Codi C.14: (AppControl) UtilsVista.java

```

1 package Utils;
2
3 import com.sun.lwuit.ComboBox;
4 import com.sun.lwuit.Container;
5 import com.sun.lwuit.Font;
6 import com.sun.lwuit.TextArea;
7 import com.sun.lwuit.TextField;
8 import com.sun.lwuit.geom.Dimension;
9 import com.sun.lwuit.layouts.BorderLayout;
10 import com.sun.lwuit.plaf.Border;
11
12 /**
13  * Classe amb funcions útils en l'àmbit de la creació de la GUI (Graphic User Interface)
14  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
15  * @version 3.1 (Maig de 2010)
16  */
17 public class UtilsVista {
18

```

```

19  /** Crea una instància de UtilsVista */
20  public UtilsVista(){
21
22  /**
23   * Crea un ComboBox amb unes determinades opcions
24   * @param opcions    Vector d'opcions a mostrar en el combobox
25   * @param labelW      Amplada del ComboBox (px)
26   * @return            ComboBox creat
27   */
28  public ComboBox crearComboBox(String[] opcions, int labelW){
29      ComboBox CB = new ComboBox(opcions);
30      CB.setPreferredW(labelW-3);
31      CB.setPreferredH(22);
32      CB.setFocusable(true);
33      CB.getStyle().setFgColor(0xFFFFFFFF);
34      CB.getStyle().setBgColor(0x444444);
35      CB.getStyle().setFgSelectionColor(0xFFFFFFFF);
36      CB.getStyle().setBgSelectionColor(0x444444);
37      CB.getStyle().setBorder(Border.createEmpty());
38      CB.getStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.STYLE_BOLD,
39          Font.SIZE_SMALL));
40      CB.getStyle().setMargin(2,2,1,1);
41      CB.getSelectedStyle().setFgColor(0xFFFFFFFF);
42      CB.getSelectedStyle().setBgColor(0x666666);
43      CB.getSelectedStyle().setFgSelectionColor(0xFFFFFFFF);
44      CB.getSelectedStyle().setBgSelectionColor(0x666666);
45      CB.getSelectedStyle().setBorder(Border.createEmpty());
46      CB.getSelectedStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.
47          STYLE_BOLD,Font.SIZE_SMALL));
48      CB.getSelectedStyle().setMargin(2,2,1,1);
49      return CB;
50  }
51
52  /**
53   * Crea un TextArea amb unes determinades característiques
54   * @param text        Text que contindrà el TextArea
55   * @param labelW      Amplada del TextArea (px)
56   * @param editable    Indica si el contingut del TextArea ha de ser editable
57   * @param focusable   Indica si el TextArea ha de poder rebre un event de focus
58   * @return            TextArea creat
59   */
60  public TextArea crearTextArea(String text, int labelW, boolean editable, boolean
61      focusable){
62      TextArea TA = new TextArea(2,2,TextArea.ANY);
63      if (focusable){ TA.getStyle().setFgColor(0xFFFFFFFF); } else { TA.getStyle().
64          setFgColor(0xCCCCFF); }
65      if (focusable){ TA.getSelectedStyle().setFgColor(0xFFFFFFFF); } else { TA.
66          getSelectedStyle().setFgColor(0xCCCCFF); }
67      TA.setPreferredW(labelW-3);
68      TA.setPreferredH(22);
69      TA.setText(text);
70      TA.setScrollSize(new Dimension(0,0));
71      TA.setEditable(editable);
72      TA.setFocusable(focusable);
73      TA.getStyle().setBgColor(0x444444);
74      TA.getStyle().setBorder(Border.createEmpty());
75      TA.getStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.STYLE_BOLD,
76          Font.SIZE_SMALL));
77      TA.getStyle().setMargin(2,2,1,1);
78      TA.getSelectedStyle().setBgColor(0x666666);
79      TA.getSelectedStyle().setBorder(Border.createEmpty());
80      TA.getSelectedStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.
81          STYLE_BOLD,Font.SIZE_SMALL));
82      TA.getSelectedStyle().setMargin(2,2,1,1);
83      return TA;
84  }
85
86  /**
87   * Crea un TextField amb unes determinades característiques
88   * @param text        Text que contindrà el TextField

```

```

82      * @param labelW      Amplada del TextField (px)
83      * @param editable    Indica si el contingut del TextField ha de ser editable
84      * @param focusable   Indica si el TextField ha de poder rebre un event de focus
85      * @param tipus       Indica el tipus de restricció del contingut del TextField
86      * @param maxsize     Indica el tamany màxim assumible pel TextField
87      * @return            TextField creat
88      */
89      public TextArea crearTextField(String text, int labelW, boolean editable, boolean
          focusable, int tipus, int maxsize){
90          TextArea TA = new TextField();
91          TA.setPreferredW(labelW-3);
92          TA.setPreferredH(22);
93          TA.setText(text);
94          TA.setConstraint(tipus);
95          if (maxsize>-1) TA.setMaxSize(maxsize);
96          TA.setScrollSize(new Dimension(0,0));
97          TA.setEditable(editable);
98          TA.setFocusable(focusable);
99          TA.getStyle().setFgColor(0xCCCCFF);
100         TA.getStyle().setBgColor(0x444444);
101         TA.getStyle().setBorder(Border.createEmpty());
102         TA.getStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.STYLE_BOLD,
            Font.SIZE_SMALL));
103         TA.getStyle().setMargin(2,2,1,1);
104         TA.getSelectedStyle().setFgColor(0xFFFFFFFF);
105         TA.getSelectedStyle().setBgColor(0x666666);
106         TA.getSelectedStyle().setBorder(Border.createEmpty());
107         TA.getSelectedStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.
            STYLE_BOLD,Font.SIZE_SMALL));
108         TA.getSelectedStyle().setMargin(2,2,1,1);
109         return TA;
110     }
111
112     /**
113     * Crea un contenedor amb un parell label<->text
114     * @param label      Etiqueta de text del parell
115     * @param text        Text que contindrà el TextArea del parell
116     * @param labelW      Amplada de cada un dels elements del parell (px)
117     * @return            Contenedor creat
118     */
119     public Container crearParell(String label, String text, int labelW){
120         Container parell = new Container(new BorderLayout());
121         parell.addComponent(BorderLayout.WEST,crearTextArea(label,labelW,false,false));
122         parell.addComponent(BorderLayout.EAST,crearTextArea(text,labelW,false,true));
123         return parell;
124     }
125
126 }

```

Codi C.15: (AppControl) FormAlarma.java

```

1  package Vista;
2
3  import Constants.CONST;
4  import Constants.LANG;
5  import Dades.Model;
6  import com.sun.lwuit.Command;
7  import com.sun.lwuit.Component;
8  import com.sun.lwuit.Container;
9  import com.sun.lwuit.Form;
10 import com.sun.lwuit.Image;
11 import com.sun.lwuit.Label;
12 import com.sun.lwuit.layouts.BorderLayout;
13
14 /**
15  * Classe que defineix els elements gràfics del formulari d'Alarma
16  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
17  * @version 3.1 (Maig de 2010)
18  */

```



```

19 public class FormAlarma extends Form {
20
21     /**
22      * Crea una instància de FormAlarma
23      * @param V          Referència a la Vista
24      * @param M          Referència al Model de dades
25      * @param idioma     Identificador de l'idioma
26      */
27     public FormAlarma(Vista V, final Model M, int idioma) throws Exception{
28         this.setLayout(new BorderLayout());
29         Container c1 = new Container(new BorderLayout());
30         Label l = new Label(Image.createImage("/alarma2.png"));
31         l.setAlignment(Component.CENTER);
32         c1.addComponent(BorderLayout.CENTER,l);
33         this.addComponent(BorderLayout.CENTER,c1);
34         this.addCommand(new Command(LANG.MENU_ATURAR[idioma], CONST.COMMAND_ATURAR));
35         this.setCommandListener(V);
36     }
37
38 }

```

### Codi C.16: (AppControl) FormConfig.java

```

1 package Vista;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import Dades.Model;
6 import Utils.UtilsVista;
7 import com.sun.lwuit.ComboBox;
8 import com.sun.lwuit.Command;
9 import com.sun.lwuit.Container;
10 import com.sun.lwuit.Form;
11 import com.sun.lwuit.Image;
12 import com.sun.lwuit.TextArea;
13 import com.sun.lwuit.TextField;
14 import com.sun.lwuit.events.ActionEvent;
15 import com.sun.lwuit.events.ActionListener;
16 import com.sun.lwuit.layouts.BorderLayout;
17 import com.sun.lwuit.layouts.BoxLayout;
18
19 /**
20  * Classe que defineix els elements gràfics del formulari de Configuració
21  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
22  * @version 3.1 (Maig de 2010)
23  */
24 public class FormConfig extends Form {
25
26     /** Objecte per a Utilitats de generació de la Vista */
27     private UtilsVista UV = new UtilsVista();
28
29     /**
30      * Crea una instància de FormConfig
31      * @param V          Referència a la Vista
32      * @param M          Referència al Model de dades
33      * @param idioma     Identificador de l'idioma
34      */
35     public FormConfig(Vista V, final Model M, int idioma) throws Exception {
36         this.setTitle(LANG.TITOL_CONFIGURACIO[idioma]);
37         this.setLayout(new BoxLayout(BoxLayout.Y_AXIS));
38         this.getTitleComponent().setIcon(Image.createImage("/logo.png"));
39         int labelW = this.getWidth()/2;
40         Container c1 = new Container(new BorderLayout());
41         c1.addComponent(BorderLayout.WEST,UV.crearTextArea(LANG.LABEL_CODIPPSTOP[idioma],labelW,false,false));
42         final TextArea TA = UV.crearTextField(M.getCodi(),labelW,true,true,TextField.ANY,12);
43         TA.addActionListener(new ActionListener(){
44             public void actionPerformed(ActionEvent evt) {

```

```

45         if (!TA.getText().equals(M.getCodi()))
46             M.setEstat(CONST.ESTAT_NOASSOCIAT);
47         M.setCodi(TA.getText());
48     }
49 });
50 c1.addComponent(BorderLayout.EAST,TA);
51 this.addComponent(c1);
52 Container c2 = new Container(new BorderLayout());
53 c2.addComponent(BorderLayout.WEST,UV.crearTextArea(LANG.LABEL_CONFIDIOMA[idioma]
54             ],labelW,false,false));
54 String opcionesIdioma[] = {
55     LANG.LABEL_CA[idioma],
56     LANG.LABEL_ES[idioma],
57     LANG.LABEL_EN[idioma]
58 };
59 final ComboBox CB1 = UV.crearComboBox(opcionesIdioma,labelW);
60 CB1.addActionListener(new ActionListener(){
61     public void actionPerformed(ActionEvent evt) {
62         M.setIdioma(CB1.getSelectedIndex());
63     }
64 });
65 CB1.setSelectedIndex(idioma);
66 c2.addComponent(BorderLayout.EAST,CB1);
67 this.addComponent(c2);
68 Container c3 = new Container(new BorderLayout());
69 c3.addComponent(BorderLayout.WEST,UV.crearTextArea(LANG.LABEL_CONFITIPUSALARMA[
70     idioma],labelW,false,false));
71 String opcionesAudio[] = {
72     LANG.LABEL_VEUHUMANA[idioma],
73     LANG.LABEL_RING[idioma],
74     LANG.LABEL_SILENCI[idioma]
75 };
76 final ComboBox CB2 = UV.crearComboBox(opcionesAudio,labelW);
77 CB2.addActionListener(new ActionListener(){
78     public void actionPerformed(ActionEvent evt) {
79         M.setTipusAlarma(CB2.getSelectedIndex());
80     }
81 });
82 CB2.setSelectedIndex(M.getTipusAlarma());
83 c3.addComponent(BorderLayout.EAST,CB2);
84 this.addComponent(c3);
85 Container c4 = new Container(new BorderLayout());
86 c4.addComponent(BorderLayout.WEST,UV.crearTextArea(LANG.LABEL_CONFITEMPSALARMA[
87     idioma],labelW,false,false));
88 String opcionesTemps[] = {
89     LANG.LABEL_T10S[idioma],
90     LANG.LABEL_T20S[idioma],
91     LANG.LABEL_T30S[idioma],
92     LANG.LABEL_TINDEFINIT[idioma]
93 };
94 final ComboBox CB3 = UV.crearComboBox(opcionesTemps,labelW);
95 CB3.addActionListener(new ActionListener(){
96     public void actionPerformed(ActionEvent evt) {
97         switch(CB3.getSelectedIndex()){
98             case 0: M.setMaxTempsAlarma(CONST.ALARMA_T10S); break;
99             case 1: M.setMaxTempsAlarma(CONST.ALARMA_T20S); break;
100             case 2: M.setMaxTempsAlarma(CONST.ALARMA_T30S); break;
101             case 3: M.setMaxTempsAlarma(CONST.ALARMA_TINDEFINIT); break;
102         }
103     }
104 });
105 switch(M.getMaxTempsAlarma()){
106     case CONST.ALARMA_T10S: CB3.setSelectedIndex(0); break;
107     case CONST.ALARMA_T20S: CB3.setSelectedIndex(1); break;
108     case CONST.ALARMA_T30S: CB3.setSelectedIndex(2); break;
109     case CONST.ALARMA_TINDEFINIT: CB3.setSelectedIndex(3); break;
110 }
111 c4.addComponent(BorderLayout.EAST,CB3);
112 this.addComponent(c4);
113 Container c5 = new Container(new BorderLayout());

```

```

112     c5.addComponent(BorderLayout.WEST,UV.crearTextArea(LANG.LABEL_CONFVOLALARMA[
113         idioma],labelW,false,false));
114     String opcionesVolum[] = {LANG.LABEL_VOL20[idioma],LANG.LABEL_VOL40[idioma],LANG.
115         LABEL_VOL60[idioma],LANG.LABEL_VOL80[idioma],LANG.LABEL_VOL100[idioma]};
116     final ComboBox CB4 = UV.crearComboBox(opcionesVolum,labelW);
117     CB4.addActionListener(new ActionListener(){
118         public void actionPerformed(ActionEvent evt) {
119             switch(CB4.getSelectedIndex()){
120                 case 0: M.setVolumAlarma(CONST.ALARMA_VOL20); break;
121                 case 1: M.setVolumAlarma(CONST.ALARMA_VOL40); break;
122                 case 2: M.setVolumAlarma(CONST.ALARMA_VOL60); break;
123                 case 3: M.setVolumAlarma(CONST.ALARMA_VOL80); break;
124                 case 4: M.setVolumAlarma(CONST.ALARMA_VOL100); break;
125             }
126         }
127     });
128     switch(M.getVolumAlarma()){
129         case CONST.ALARMA_VOL20: CB4.setSelectedIndex(0); break;
130         case CONST.ALARMA_VOL40: CB4.setSelectedIndex(1); break;
131         case CONST.ALARMA_VOL60: CB4.setSelectedIndex(2); break;
132         case CONST.ALARMA_VOL80: CB4.setSelectedIndex(3); break;
133         case CONST.ALARMA_VOL100: CB4.setSelectedIndex(4); break;
134     }
135     c5.addComponent(BorderLayout.EAST,CB4);
136     this.addComponent(c5);
137     this.addCommand(new Command(LANG.MENU_TORNAR[idioma], CONST.COMMAND_TORNAR));
138     this.setCommandListener(V);
139 }

```

### Codi C.17: (AppControl) FormHistorial.java

```

1 package Vista;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import Dades.Model;
6 import com.sun.lwuit.Command;
7 import com.sun.lwuit.Component;
8 import com.sun.lwuit.Container;
9 import com.sun.lwuit.Font;
10 import com.sun.lwuit.Form;
11 import com.sun.lwuit.Image;
12 import com.sun.lwuit.Label;
13 import com.sun.lwuit.List;
14 import com.sun.lwuit.layouts.BorderLayout;
15 import com.sun.lwuit.list.ListCellRenderer;
16 import java.util.Vector;
17
18 /**
19  * Classe que defineix els elements gràfics del formulari d'Historial d'Alarmes
20  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
21  * @version 3.1 (Maig de 2010)
22  */
23 public class FormHistorial extends Form {
24
25     /**
26      * Crea una instància de FormHistorial
27      * @param V Referència a la Vista
28      * @param M Referència al Model de dades
29      * @param idioma Identificador de l'idioma
30      */
31     public FormHistorial(Vista V, Model M, int idioma) throws Exception {
32         this.setTitle(LANG.TITOL_HISTORIALALARMES[idioma]);
33         this.setLayout(new BorderLayout());
34         this.setScrollable(false);
35         this.getTitleComponent().setIcon(Image.createImage("/logo.png"));
36         Vector vec = M.getHistorialTXT();

```

```

37         if (!vec.isEmpty()){
38             List L = new List(vec);
39             L.setListCellRenderer(new LlistaRenderer());
40             this.addComponent(BorderLayout.CENTER,L);
41         } else {
42             Label l = new Label(LANG.TEXT_NOALARMES[M.getIdioma()]);
43             l.getStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.
44                 STYLE_ITALIC,Font.SIZE_SMALL));
45             this.addComponent(BorderLayout.NORTH,l);
46         }
47         this.addCommand(new Command(LANG.MENU_TORNAR[idioma], CONST.COMMAND_TORNAR));
48         if (!vec.isEmpty())
49             this.addCommand(new Command(LANG.MENU_ESBORRAR[idioma], CONST.
50                 COMMAND_ESBORRARDIALOG));
51         this.setCommandListener(V);
52     }
53
54     /**
55     * Subclasse que defineix l'aspecte gràfic de la Llista
56     */
57     class LlistaRenderer extends Container implements ListCellRenderer {
58
59         /** Etiqueta de background de cada element de la llista */
60         private Label imatge = new Label();
61         /** Etiqueta de text de cada element de la llista */
62         private Label alarma = new Label();
63         /** Etiqueta de focus de cada element de la llista */
64         private Label focus = new Label();
65         /** Icona de cada element de la llista */
66         private Image icon = Image.createImage("/alarma.png");
67
68         /**
69         * Components de cada element de la Llista
70         */
71         public LlistaRenderer() throws Exception {
72             setLayout(new BorderLayout());
73             imatge.getStyle().setBgColor(0x888888);
74             imatge.getStyle().setMargin(1,1,0,0);
75             imatge.getStyle().setBgTransparency(75);
76             alarma.setPreferredH(22);
77             alarma.getStyle().setBgColor(0x888888);
78             alarma.getStyle().setMargin(1,1,0,0);
79             alarma.getStyle().setBgTransparency(75);
80             addComponent(BorderLayout.WEST, imatge);
81             addComponent(BorderLayout.CENTER, alarma);
82             focus.setFocus(true);
83         }
84
85         /**
86         * Definició del valor de cada component de cada element de la llista
87         * @param list      Llista a aplicar Render
88         * @param value      Valor de l'element de la llista del Render actual
89         * @param index      Index de l'element de la llista del Render actual
90         * @param isSelected Indica si l'element està seleccionat o no
91         * @return           Component sobre el qual s'ha fet Render
92         */
93         public Component getListCellRendererComponent(List list, Object value, int index,
94             boolean isSelected) {
95             imatge.setIcon(icon);
96             alarma.setText(value.toString());
97             return this;
98         }
99
100         /**
101         * Definició de l'estil de focus d'un element de la llista
102         * @param list      Llista a aplicar Render
103         * @return           Component enfocat sobre el qual s'ha fet render
104         */
105         public Component getListFocusComponent(List list) {
106             focus.getStyle().setBgColor(0x444444);
107         }
108     }

```

```

104         focus.setPreferredH(22);
105         focus.getStyle().setMargin(1,1,0,0);
106         return focus;
107     }
108 }
109
110 }

```

Codi C.18: (AppControl) FormMenu.java

```

1 package Vista;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import com.sun.lwuit.Button;
6 import com.sun.lwuit.Command;
7 import com.sun.lwuit.Component;
8 import com.sun.lwuit.Container;
9 import com.sun.lwuit.Font;
10 import com.sun.lwuit.Form;
11 import com.sun.lwuit.Image;
12 import com.sun.lwuit.Label;
13 import com.sun.lwuit.events.ActionEvent;
14 import com.sun.lwuit.events.ActionListener;
15 import com.sun.lwuit.events.FocusListener;
16 import com.sun.lwuit.layouts.BorderLayout;
17 import com.sun.lwuit.layouts.GridLayout;
18 import com.sun.lwuit.plaf.Border;
19
20 /**
21  * Classe que defineix els elements gràfics del formulari de Menú Principal
22  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
23  * @version 3.1 (Maig de 2010)
24  */
25 public class FormMenu extends Form {
26
27     /** Referència a la Vista */
28     private Vista V;
29     /** Etiqueta de text del menú */
30     private Label lm;
31
32     /**
33      * Crea una instància de Form Menu
34      * @param V Referència a la Vista
35      * @param idioma Identificador de l'idioma
36      */
37     public FormMenu(Vista V, int idioma) throws Exception {
38         this.V = V;
39         this.setLayout(new BorderLayout());
40         Container c1 = new Container(new BorderLayout());
41         c1.setPreferredH((this.getHeight()-160)/2-15);
42         Label l = new Label(Image.createImage("/titol.png"));
43         l.setAlignment(Component.CENTER);
44         c1.addComponent(BorderLayout.CENTER,l);
45         this.addComponent(BorderLayout.NORTH,c1);
46         Container c2 = new Container(new GridLayout(2,3));
47         c2.setPreferredH(160);
48         final String[] textMenu = {
49             LANG.TITOL_RESUM[idioma],
50             LANG.TITOL_BUSCARTERMINALS[idioma],
51             LANG.TITOL_MODIFICARSENSIBILITAT[idioma],
52             LANG.TITOL_CONFIGURACIO[idioma],
53             LANG.TITOL_HISTORIALALARMES[idioma],
54             LANG.TITOL_ROTARTERMINALS[idioma],
55             //LANG.TITOL_TRANSFERIRHISTORIAL[idioma]
56         };
57         final int[] destiMenu = {
58             CONST.FORM_RESUM,
59             CONST.FORM_BUSCARTERMINALS,

```

```

60         CONST.FORM_MODIFICARSENSIBILITAT,
61         CONST.FORM_CONFIGURACIO,
62         CONST.FORM_HISTORIALALARMES,
63         CONST.FORM_ROTARTERMINALS,
64         //CONST.FORM_TRANSFERIRHISTORIAL
65     };
66     for (int k=0;k<6;k++)
67         c2.addComponent(crearBotoMenu(k+1,textMenu[k],destiMenu[k]));
68     this.addComponent(BorderLayout.CENTER,c2);
69     Container c3 = new Container(new BorderLayout());
70     c3.setPreferredH((this.getHeight()-160)/2-15);
71     lm = new Label();
72     lm.getStyle().setFont(Font.createSystemFont(Font.FACE_SYSTEM,Font.STYLE_BOLD,
73         Font.SIZE_LARGE));
74     lm.setAlignment(Component.CENTER);
75     c3.addComponent(BorderLayout.CENTER,lm);
76     this.addComponent(BorderLayout.SOUTH,c3);
77     this.addCommand(new Command(LANG.MENU_SORTIR[idioma], CONST.COMMAND_SORTIRDIALOG
78         ));
79     this.setCommandListener(V);
80 }
81 /**
82  * Crea un botó per al Men Principal, amb un text i destí determinats
83  * @param n            Identificador numèric del botó a crear
84  * @param text         Text a aplicar a l'etiqueta del menú en l'event FocusGained
85  * @param formDesti     Identificador del formulari destí del botó
86  * @return             Botó creat
87  */
88 private Button crearBotoMenu(int n, final String text, final int formDesti) throws
89     Exception{
90     Button b = null;
91     b = new Button(Image.createImage("/b" + n + ".png"));
92     b.setRolloverIcon(Image.createImage("/b" + n + "on.png"));
93     b.setAlignment(Label.CENTER);
94     b.getStyle().setBgColor(0x000000);
95     b.getStyle().setBgSelectionColor(0x000000);
96     b.getStyle().setMargin(0,0,0,0);
97     b.getStyle().setPadding(0,0,0,0);
98     b.getStyle().setBorder(Border.createEmpty());
99     b.getSelectedStyle().setBorder(Border.createEmpty());
100    b.getPressedStyle().setBgColor(0x000000);
101    b.addFocusListener(new FocusListener(){
102        public void focusGained(Component evt) {
103            lm.setText(text);
104        }
105        public void focusLost(Component arg0) {}
106    });
107    b.addActionListener(new ActionListener() {
108        public void actionPerformed(ActionEvent evt) {
109            V.mostrarFormulari(formDesti);
110        }
111    });
112    return b;
113 }

```

### Codi C.19: (AppControl) FormResum.java

```

1 package Vista;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import Dades.Model;
6 import Utils.UtilsVista;
7 import com.sun.lwuit.Command;
8 import com.sun.lwuit.Form;
9 import com.sun.lwuit.Image;

```

```

10 import com.sun.lwuit.layouts.BoxLayout;
11
12 /**
13  * Classe que defineix els elements gràfics del formulari de Resum
14  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
15  * @version 3.1 (Maig de 2010)
16  */
17 public class FormResum extends Form {
18
19     /** Objecte per a Utilitats de generació de la Vista */
20     private UtilsVista UV = new UtilsVista();
21
22     /**
23      * Crea una instància de FormResum
24      * @param V Referència a la Vista
25      * @param M Referència al Model de dades
26      * @param idioma Identificador de l'idioma
27      */
28     public FormResum(Vista V, Model M, int idioma) throws Exception {
29         this.setTitle(LANG.TITOL_RESUM[idioma]);
30         this.setLayout(new BoxLayout(BoxLayout.Y_AXIS));
31         this.getTitleComponent().setIcon(Image.createImage("/logo.png"));
32         int labelW = this.getWidth()/2;
33         if (M.getCodi().equals("")){
34             this.addComponent(UV.crearParell(LANG.LABEL_CODIPPSTOP[idioma], "--", labelW))
35             ;
36         } else {
37             this.addComponent(UV.crearParell(LANG.LABEL_CODIPPSTOP[idioma], M.getCodi(),
38                 labelW));
39         }
40         if (M.getMacMovil().equals("")){
41             this.addComponent(UV.crearParell(LANG.LABEL_CODIHISTORIAL[idioma], "--",
42                 labelW));
43         } else {
44             this.addComponent(UV.crearParell(LANG.LABEL_CODIHISTORIAL[idioma], M.
45                 getMacMovil(), labelW));
46         }
47         if (M.getEstatBT()==CONST.ESTAT_ACTIVAT){
48             this.addComponent(UV.crearParell(LANG.LABEL_ESTATBLUETOOTH[idioma], LANG.
49                 LABEL_ACTIVAT[idioma], labelW));
50         } else {
51             this.addComponent(UV.crearParell(LANG.LABEL_ESTATBLUETOOTH[idioma], LANG.
52                 LABEL_DESACTIVAT[idioma], labelW));
53         }
54         if (M.getEstat()==CONST.ESTAT_ASSOCIAT){
55             this.addComponent(UV.crearParell(LANG.LABEL_ESTATSERVEI[idioma], LANG.
56                 LABEL_ASSOCIAT[idioma], labelW));
57         } else {
58             this.addComponent(UV.crearParell(LANG.LABEL_ESTATSERVEI[idioma], LANG.
59                 LABEL_NOASSOCIAT[idioma], labelW));
60         }
61         //if (M.getBateria()==CONST.NODEF){
62         //    this.addComponent(UV.crearParell(LANG.LABEL_ESTATBATERIA[idioma], "--", labelW
63         //));
64         //} else {
65         //    this.addComponent(UV.crearParell(LANG.LABEL_ESTATBATERIA[idioma], String.
66         //        valueOf(M.getBateria())+"%", labelW));
67         //}
68         if (M.getOrdre()==CONST.NODEF){
69             this.addComponent(UV.crearParell(LANG.LABEL_ORDRETERMINAL[idioma], "--",
70                 labelW));
71         } else {
72             this.addComponent(UV.crearParell(LANG.LABEL_ORDRETERMINAL[idioma], String.
73                 valueOf(M.getOrdre()), labelW));
74         }
75         String tSen[] = {
76             LANG.LABEL_BAIXA[idioma],
77             LANG.LABEL_ALTA[idioma],
78         };
79         if (M.getSensibilitat()==CONST.NODEF){

```

```

68         this.addComponent(UV.crearParell(LANG.LABEL_SENSIBILITAT[idioma], "--", labelW
        ));
69     } else {
70         this.addComponent(UV.crearParell(LANG.LABEL_SENSIBILITAT[idioma], tSen[M.
        getSensibilitat()], labelW));
71     }
72     String s = M.getDarreraAlarmaDif();
73     if (s==null){
74         this.addComponent(UV.crearParell(LANG.LABEL_DARRERAALARMA[idioma], "--",
        labelW));
75     } else {
76         this.addComponent(UV.crearParell(LANG.LABEL_DARRERAALARMA[idioma], M.
        getDarreraAlarma(), labelW));
77         this.addComponent(UV.crearParell("", M.getDarreraAlarmaDif(), labelW));
78     }
79     String t = null;
80     if (M.getEstat()==CONST.ESTAT_ASSOCIAT){
81         switch (M.getFlag()){
82             case CONST.FLAG_ALARMA: t = LANG.TEXT_FLAGALARMA[idioma]; break;
83             case CONST.FLAG_BUSCARTERMINALS: t = LANG.TEXT_FLAGBUSCARTERMINALS[
        idioma]; break;
84             case CONST.FLAG_MODIFICARSENSIBILITAT: t = (LANG.
        TEXT_FLAGCANVIARSENSIBILITAT[idioma]+"("+M.getSensibilitatTX()+")");
        break;
85             case CONST.FLAG_ROTARTERMINALS: t = LANG.TEXT_FLAGROTARTERMINALS[idioma
        ]; break;
86         }
87     } else {
88         t = LANG.TEXT_FLAGCOMPROVAR[idioma];
89     }
90     this.addComponent(UV.crearParell(LANG.LABEL_PROXIMAACCIO[idioma], t, labelW));
91     this.addCommand(new Command(LANG.MENU_TORNAR[idioma], CONST.COMMAND_TORNAR));
92     this.setCommandListener(V);
93 }
94
95 }

```

Codi C.20: (AppControl) FormSplash.java

```

1 package Vista;
2
3 import com.sun.lwuit.Container;
4 import com.sun.lwuit.Form;
5 import com.sun.lwuit.Image;
6 import com.sun.lwuit.animations.CommonTransitions;
7 import com.sun.lwuit.layouts.BorderLayout;
8
9 /**
10  * Classe que defineix els elements gràfics del formulari de Splash
11  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
12  * @version 3.1 (Maig de 2010)
13  */
14 public class FormSplash extends Form {
15
16     /**
17      * Crea una instància de FormSplash
18      */
19     public FormSplash() throws Exception{
20         this.setLayout(new BorderLayout());
21         Container c = new Container();
22         this.getStyle().setBgImage(Image.createImage("/splash.png"));
23         this.addComponent(BorderLayout.CENTER, c);
24         this.setTransitionOutAnimator(CommonTransitions.createEmpty());
25     }
26
27 }

```

Codi C.21: (AppControl) Vista.java



```

1 package Vista;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import Dades.Model;
6 import Main.PPStop;
7
8 import Utils.UtilsVista;
9 import com.sun.lwuit.*;
10 import com.sun.lwuit.events.*;
11 import com.sun.lwuit.animations.CommonTransitions;
12 import com.sun.lwuit.layouts.BorderLayout;
13 import com.sun.lwuit.layouts.BoxLayout;
14 import com.sun.lwuit.plaf.Border;
15 import com.sun.lwuit.plaf.UIManager;
16 import com.sun.lwuit.util.Resources;
17 import java.io.IOException;
18 import java.util.Timer;
19 import java.util.TimerTask;
20
21
22 /**
23  * Classe que conté tota la gestió de la GUI (Graphic User Interface). Actua com a Vista
24  * dins del patró MVC (Model-View-Controller)
25  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
26  * @version 3.1 (Maig de 2010)
27  */
28 public class Vista implements ActionListener{
29
30     /** Referència al MIDlet */
31     private PPStop P;
32     /** Referència al Model de dades */
33     private Model M;
34     /** Timer par a timeouts */
35     private Timer T;
36     /** Identificador del formulari que s'està mostrant */
37     private int formActual;
38     /** Identificador del formulari mostrat anteriorment al que s'està mostrant */
39     private int formAnterior;
40     /** Objecte Dialog */
41     private Dialog D;
42     /** Command Button esquerre del Dialog */
43     private Command cmd1;
44     /** Command Button dret del Dialog */
45     private Command cmd2;
46     /** ComboBox del Dialog per a selecció de la nova sensibilitat a TX */
47     private ComboBox CB_NovaSensibilitat;
48     /** Objecte per a Utilitats de generació de la Vista */
49     private UtilsVista UV = new UtilsVista();
50
51     /**
52     * Crea una instància de Vista
53     * @param P Referència al MIDlet
54     * @param M Referència al Model de dades
55     */
56     public Vista(PPStop P, Model M){
57         this.P = P;
58         this.M = M;
59         Display.init(this.P);
60         setDefaultStyles("/ppstop.res","PPStop");
61         initComponents();
62     }
63
64     /**
65     * Mostra per pantalla un determinat formulari
66     * @param f Identificador del formulari a mostrar
67     */
68     public void mostrarFormulari(int f){
69         try {

```

```

69     if (f == CONST.FORM_ANTERIOR)
70         f = formAnterior;
71     if (f != CONST.FORM_SPLASH)
72         T.cancel();
73     switch (f) {
74         case CONST.FORM_SPLASH:
75             M.setRefresh(false);
76             new FormSplash().show();
77             formActual = f;
78             break;
79         case CONST.FORM_MENU:
80             M.setRefresh(false);
81             new FormMenu(this,M.getIdioma()).show();
82             formActual = f;
83             break;
84         case CONST.FORM_RESUM:
85             M.setRefresh(true);
86             new FormResum(this,M,M.getIdioma()).show();
87             T = new Timer();
88             T.schedule(new endTimerRefresc(),60*1000);
89             formActual = f;
90             break;
91         case CONST.FORM_BUSCARTERMINALS:
92             M.setRefresh(false);
93             if (M.getEstatBT()==CONST.ESTAT_DESACTIVAT){
94                 mostrarDialog(Dialog.TYPE_ERROR,LANG.TEXT_NOBTACTIU[M.getIdioma()
95                     ],null,null,CONST.TEMPS_DIALOG,70);
96             } else if (M.getEstat()==CONST.ESTAT_NOASSOCIAT){
97                 mostrarDialog(Dialog.TYPE_ERROR,LANG.TEXT_NOASSOCIAT[M.getIdioma()
98                     ],cmd1,cmd2,CONST.TEMPS_DIALOG,20);
99             } else {
100                 if (M.getFlag()==CONST.FLAG_BUSCARTERMINALS){
101                     cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.COMMAND_CANCELARACCIO);
102                     cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.COMMAND_RES);
103                     mostrarDialog(Dialog.TYPE_CONFIRMATION,LANG.TEXT_CANCELARACCIO[M.getIdioma()],cmd1,cmd2,0,70);
104                 } else {
105                     cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.COMMAND_BUSCARTERMINALS);
106                     cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.COMMAND_RES);
107                     mostrarDialog(Dialog.TYPE_CONFIRMATION,LANG.TEXT_CONFIRMACIOBUSCARTERMINALS[M.getIdioma()],cmd1,cmd2,0,130);
108                 }
109             }
110             break;
111         case CONST.FORM_MODIFICARSENSIBILITAT:
112             M.setRefresh(false);
113             if (M.getEstatBT()==CONST.ESTAT_DESACTIVAT){
114                 mostrarDialog(Dialog.TYPE_ERROR,LANG.TEXT_NOBTACTIU[M.getIdioma()
115                     ],null,null,CONST.TEMPS_DIALOG,70);
116             } else if (M.getEstat()==CONST.ESTAT_NOASSOCIAT){
117                 mostrarDialog(Dialog.TYPE_ERROR,LANG.TEXT_NOASSOCIAT[M.getIdioma()
118                     ],cmd1,cmd2,CONST.TEMPS_DIALOG,20);
119             } else {
120                 if (M.getFlag()==CONST.FLAG_MODIFICARSENSIBILITAT){
121                     cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.COMMAND_CANCELARACCIO);
122                     cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.COMMAND_RES);
123                     mostrarDialog(Dialog.TYPE_CONFIRMATION,LANG.TEXT_CANCELARACCIO[M.getIdioma()],cmd1,cmd2,0,70);
124                 } else {
125                     cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.COMMAND_MODIFICARSENSIBILITAT);
126                     cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.COMMAND_RES);
127                 }
128             }
129         }
130     }

```

```

123         mostrarDialogSensibilitat(cmd1,cmd2);
124     }
125 }
126 break;
127 case CONST.FORM_CONFIGURACIO:
128     M.setRefresh(false);
129     new FormConfig(this,M,M.getIdioma()).show();
130     formActual = f;
131     break;
132 case CONST.FORM_HISTORIALALARMES:
133     M.setRefresh(true);
134     new FormHistorial(this,M,M.getIdioma()).show();
135     formActual = f;
136     break;
137 case CONST.FORM_TRANSFERIRHISTORIAL:
138     M.setRefresh(false);
139     if (M.getEstatBT()==CONST.ESTAT_DESACTIVAT){
140         mostrarDialog(Dialog.TYPE_ERROR,LANG.TEXT_NOBTACTIU[M.getIdioma()
141             ],null,null,CONST.TEMPS_DIALOG,50);
142     } else {
143         cmd1 = new Command(LANG.MENU_OK[M.getIdioma()],CONST.COMMAND_RES
144             );
145         mostrarDialog(Dialog.TYPE_INFO,LANG.TEXT_TRANSFERIRHISTORIAL[M.
146             getIdioma()] + M.getMacMovil(),cmd1,null,0,135);
147     }
148     break;
149 case CONST.FORM_ROTAR TERMINALS:
150     M.setRefresh(false);
151     if (M.getEstatBT()==CONST.ESTAT_DESACTIVAT){
152         mostrarDialog(Dialog.TYPE_ERROR,LANG.TEXT_NOBTACTIU[M.getIdioma()
153             ],null,null,CONST.TEMPS_DIALOG,50);
154     } else if (M.getEstat()==CONST.ESTAT_NOASSOCIAT){
155         mostrarDialog(Dialog.TYPE_ERROR,LANG.TEXT_NOASSOCIAT[M.getIdioma()
156             ],cmd1,cmd2,CONST.TEMPS_DIALOG,20);
157     } else {
158         if (M.getFlag()==CONST.FLAG_ROTAR TERMINALS){
159             cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.
160                 COMMAND_CANCELARACCIO);
161             cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.
162                 COMMAND_RES);
163             mostrarDialog(Dialog.TYPE_CONFIRMATION,LANG.
164                 TEXT_CANCELARACCIO[M.getIdioma()],cmd1,cmd2,0,70);
165         } else {
166             cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.
167                 COMMAND_ROTAR TERMINALS);
168             cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.
169                 COMMAND_RES);
170             mostrarDialog(Dialog.TYPE_CONFIRMATION,LANG.
171                 TEXT_CONFIRMACIOROTAR TERMINALS[M.getIdioma()],cmd1,cmd2
172                 ,0,130);
173         }
174     }
175     break;
176 case CONST.FORM_ALARMA:
177     M.setRefresh(false);
178     formAnterior = formActual;
179     new FormAlarma(this,M,M.getIdioma()).show();
180     formActual = f;
181     break;
182 }
183 formActual = f;
184 } catch (Exception ex) {
185     ex.printStackTrace();
186 }
187 }
188 }
189
190 /**
191  * Defineix els estils per defecte dels components de la vista: carrega un tema d'un
192  * determinat recurs
193  * @param recurs Recurs en el que es troba el tema a aplicar

```

```

180     * @param tema        Tema a aplicar
181     */
182     private void setDefaultStyles(String recurs, String tema){
183         try {
184             Resources r = Resources.open(recurs);
185             UIManager.getInstance().setThemeProps(r.getTheme(tema));
186         } catch (IOException ex) {
187             ex.printStackTrace();
188         }
189         UIManager.getInstance().getLookAndFeel().setDefaultFormTransitionOut(
190             CommonTransitions.createFade(1000));
191     }
192
193     /**
194     * Final del Timer de Splash Screen: Mostrar Men Principal
195     */
196     private class endSplashScreen extends TimerTask {
197         public void run(){
198             mostrarFormulari(CONST.FORM_MENU);
199             P.PlayAudio("/loading.mp3","audio/mpeg");
200         }
201     }
202
203     /**
204     * Final del Timer de Refresc: Refrescar formulari
205     */
206     private class endTimerRefresc extends TimerTask {
207         public void run(){
208             update();
209         }
210     }
211
212     /**
213     * Mostra per pantalla un Dialog d'un cert tipus i amb una certa configuració
214     * @param tipus        Tipus de Dialog
215     * @param text        Text que ha de mostrar el Dialog
216     * @param cmd1        Command Button de l'esquerre del Dialog
217     * @param cmd2        Command Button de la dreta del Dialog
218     * @param timeout      Timeout del Dialog (ms)
219     * @param height       Alçada del Dialog (px)
220     */
221     public void mostrarDialog(int tipus, String text, Command cmd1, Command cmd2, long
222         timeout, int height){
223         D = new Dialog();
224         D.setDialogType(tipus);
225         Container c = new Container(new BorderLayout());
226         TextArea ta = new TextArea(2,2,TextArea.UNEDITABLE);
227         ta.setPreferredW(210);
228         ta.setPreferredH(height);
229         ta.setText(text);
230         ta.setFocusable(false);
231         ta.setEditable(false);
232         ta.getStyle().setBgTransparency(0);
233         ta.getStyle().setFgColor(0xCCCCFF);
234         ta.getStyle().setBorder(Border.createEmpty());
235         c.addComponent(BorderLayout.CENTER,ta);
236         D.addComponent(c);
237         if (cmd1!=null) D.addCommand(cmd1);
238         if (cmd2!=null) D.addCommand(cmd2);
239         if (timeout>0) D.setTimeout(timeout);
240         D.setCommandListener(this);
241         D.setAutoDispose(false);
242         D.showPacked(BorderLayout.CENTER, true);
243     }
244
245     /**
246     * Mostra per pantalla el Dialog de Nova Sensibilitat
247     * @param cmd1        Command Button de l'esquerra del Dialog
248     * @param cmd2        Command Button de la dreta del Dialog
249     */

```

```

248 public void mostrarDialogSensibilitat(Command cmd1, Command cmd2){
249     D = new Dialog();
250     D.setDialogType(Dialog.TYPE_CONFIRMATION);
251     D.setFocusable(true);
252     Container c = new Container(new BoxLayout(BoxLayout.Y_AXIS));
253     TextArea ta = new TextArea(2,2,TextArea.UNEDITABLE);
254     ta.setPreferredW(210);
255     ta.setPreferredH(140);
256     ta.setText(LANG.TEXT_CONFIRMACIOMODIFICARSENSIBILITAT[M.getIdioma()]);
257     ta.setFocusable(false);
258     ta.setEditable(false);
259     ta.getStyle().setBgTransparency(0);
260     ta.getStyle().setFgColor(0xCCCCFF);
261     ta.getStyle().setBorder(Border.createEmpty());
262     c.addComponent(ta);
263     String opcionsSensibilitat[] = {
264         LANG.LABEL_BAIXA[M.getIdioma()],
265         LANG.LABEL_ALTA[M.getIdioma()]
266     };
267     CB_NovaSensibilitat = UV.crearComboBox(opcionsSensibilitat,150);
268     CB_NovaSensibilitat.setFocus(true);
269     c.addComponent(CB_NovaSensibilitat);
270     D.addComponent(c);
271     if (cmd1!=null) D.addCommand(cmd1);
272     if (cmd2!=null) D.addCommand(cmd2);
273     D.setCommandListener(this);
274     D.setAutoDispose(false);
275     D.showPacked(BorderLayout.CENTER, true);
276 }
277
278 /**
279  * Inicialitza la Graphic User Interface (GUI)
280  */
281 private void initComponents(){
282     mostrarFormulari(CONST.FORM_SPLASH);
283     T = new Timer();
284     T.schedule(new endSplashScreen(),CONST.TEMPS_SPLASHSCREEN);
285 }
286
287 /**
288  * Actualitza la Vista
289  */
290 public void update(){
291     if (M.getRefresh())
292         mostrarFormulari(formActual);
293 }
294
295 /**
296  * Listener d'events de Command Button: Realitza es accions pertinents
297  * @param evt      Event generat
298  */
299 public void actionPerformed(ActionEvent evt) {
300     Command cmd = evt.getCommand();
301     switch (cmd.getId()) {
302         case CONST.COMMAND_RES:
303             D.dispose();
304             update();
305             break;
306         case CONST.COMMAND_ESBORRARDIALOG:
307             cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.COMMAND_ESBORRAR);
308             cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.COMMAND_RES);
309             mostrarDialog(Dialog.TYPE_CONFIRMATION,LANG.TEXT_RESETHISTORIAL[M.getIdioma()],cmd1,cmd2,0,50);
310             break;
311         case CONST.COMMAND_ESBORRAR:
312             M.resetHistorial();
313             break;
314         case CONST.COMMAND_CANCELARACCIO:
315             M.setFlag(CONST.FLAG_ALARMA);

```

```

316         D.dispose();
317         break;
318     case CONST.COMMAND_BUSCARTERMINALS:
319         M.setFlag(CONST.FLAG_BUSCARTERMINALS);
320         D.dispose();
321         break;
322     case CONST.COMMAND_MODIFICARSENSIBILITAT:
323         M.setSensibilitatTX(CB_NovaSensibilitat.getSelectedIndex());
324         M.setFlag(CONST.FLAG_MODIFICARSENSIBILITAT);
325         D.dispose();
326         break;
327     case CONST.COMMAND_ROTARTERMINALS:
328         M.setFlag(CONST.FLAG_ROTARTERMINALS);
329         D.dispose();
330         break;
331     case CONST.COMMAND_ATURAR:
332         P.StopAlarma();
333         break;
334     case CONST.COMMAND_TORNAR:
335         mostrarFormulari(CONST.FORM_MENU);
336         break;
337     case CONST.COMMAND_SORTIRDIALOG:
338         cmd1 = new Command(LANG.MENU_SI[M.getIdioma()],CONST.COMMAND_SORTIR)
339             ;
340         cmd2 = new Command(LANG.MENU_NO[M.getIdioma()],CONST.COMMAND_RES);
341         mostrarDialog(Dialog.TYPE_CONFIRMATION,LANG.TEXT_SORTIR[M.getIdioma()
342             ],cmd1,cmd2,0,20);
343         break;
344     case CONST.COMMAND_SORTIR:
345         P.exitMIDlet();
346         break;
347 }
348 }

```

## C.4 Guia de l'Usuari

# Guia d'Usuari

Versió 3.1



Maig de 2010



# Contingut

<b>1</b>	<b>Posada en funcionament</b>	<b>1</b>
1.1	Descàrrega de l'aplicació . . . . .	1
1.2	Instal·lació de l'aplicació . . . . .	1
1.3	Iniciar l'aplicació . . . . .	1
1.4	Opcions disponibles . . . . .	1
1.5	Configuració inicial . . . . .	2
1.6	Associació al servei PPStop . . . . .	2
<b>2</b>	<b>Paràmetres de configuració</b>	<b>3</b>
2.1	Codi PPStop . . . . .	3
2.2	Idioma de l'interfície . . . . .	3
2.3	Tipus d'alarma . . . . .	3
2.4	Temps màxim d'alarma . . . . .	3
2.5	Volum de l'alarma . . . . .	3
<b>3</b>	<b>Paràmetres de resum</b>	<b>4</b>
3.1	Codi PPStop . . . . .	4
3.2	Codi Historial . . . . .	4
3.3	Estat de Bluetooth . . . . .	4
3.4	Estat de Servei . . . . .	4
3.5	Ordre del terminals . . . . .	4
3.6	Sensibilitat del sensor . . . . .	4
3.7	Darrera alarma . . . . .	4
3.8	Pròxima acció . . . . .	4
<b>4</b>	<b>Alarma</b>	<b>5</b>
4.1	Alarma . . . . .	5
4.2	Historial d'alarmes . . . . .	5
4.3	Esborrar l'historial . . . . .	5
<b>5</b>	<b>Canvis sobre el dispositiu PPStop</b>	<b>7</b>
5.1	Associar nous terminals al servei . . . . .	7
5.2	Modificar la sensibilitat del sensor . . . . .	7
5.3	Rotar els terminals associats . . . . .	8
5.4	Cancel·lar una acció sobre el dispositiu . . . . .	9
<b>6</b>	<b>Notes sobre l'aplicació</b>	<b>10</b>
6.1	Requisits mínims . . . . .	10
6.2	Changelog . . . . .	10
6.3	Bugs i suggeriments . . . . .	10



## Posada en funcionament

### 1.1 Descàrrega de l'aplicació

Per a descarregar l'aplicació cal enviar un SMS amb la paraula "**ppstop**" al **44505**. Automàticament rebreu l'aplicació al vostre terminal.

### 1.2 Instal·lació de l'aplicació

Una vegada rebuda l'aplicació, la seva instal·lació s'inicia de forma automàtica. Cal acceptar totes les preguntes i finalitzar el procés.

En cas que prèviament tinguéssiu instal·lada una versió anterior, s'ha de seleccionar **NO** quan es pregunta si es volen mantenir les dades de la versió anterior.

### 1.3 Iniciar l'aplicació

Una vegada instal·lada l'aplicació, us apareixerà una nova icona a la graella d'aplicacions del terminal. Aquesta icona dóna accés a l'aplicació.



**PPSTOP**

S'ha de seleccionar **SI** quan es pregunta si es vol permetre a l'aplicació utilitzar les aplicacions de connectivitat.

### 1.4 Opcions disponibles

Una vegada iniciada l'aplicació apareix el **menú principal**.



Les opcions disponibles són:



**Resum**



**Buscar Terminals**



**Modificar Sensibilitat**



**Configuració**



**Historial d'Alarmes**



**Rotar Terminals**



## 1.5 Configuració inicial

Per a començar a utilitzar l'aplicació, s'ha de configurar l'aplicació amb el **codi PPSTOP** proporcionat amb el dispositiu **PPSTOP**. Per a fer-ho, cal anar a la **pantalla de configuració** i introduir el codi a la casella corresponent. Un **codi PPSTOP** està format per 12 caràcters (0-9;A-F) i és indiferent introduir-lo en majúscula que en minúscula.



## 1.6 Associació al servei PPStop

Una vegada feta la configuració inicial, cal que el terminal s'associï al servei. Per a fer-ho, s'han de seguir els següents passos:

- 1 Apropar el terminal al dispositiu PPSTOP.
- 2 Activar voluntàriament el sensor d'humitat del dispositiu.
- 3 Esperar l'aparició d'un missatge al terminal indicant que el terminal s'ha associat al servei.

Un cop s'ha fet l'associació al servei, **PPSTOP** ja està a punt per a ser utilitzat amb normalitat.



Es pot comprovar l'estat actual d'associació al servei a través del camp **Estat Servei** de la **Pantalla de Resum**.



## Paràmetres de configuració

### 2.1 Codi PPStop

Codi de 12 caràcters (0-9;A-F) que identifica el dispositiu **PPSTOP**.

Cada vegada que es modifica aquest paràmetre, l'aplicació passa a estat **NO ASSOCIAT** i no torna a passar a estat **ASSOCIAT** fins que rep un avís d'alarma (cas de que el terminal ja hagués estat prèviament associat al servei **PPSTOP**) o fins que el dispositiu PPStop associi el terminal.

### 2.2 Idioma de l'interfície

Defineix l'**idioma** de l'aplicació.

Els idiomes disponibles per a aquesta versió són:

<b>Català</b>	Estableix l'idioma com a català
<b>Castellà</b>	Estableix l'idioma com a castellà
<b>Anglès</b>	Estableix l'idioma com a anglès

### 2.3 Tipus d'alarma

Defineix el **tipus d'alarma** de l'aplicació.

Els tipus d'alarma disponibles per a aquesta versió són:

<b>Veu humana</b>	Estableix com a timbre d'alarma una veu humana
<b>Ring</b>	Estableix com a timbre d'alarma un RING
<b>Silenci</b>	Estableix una alarma únicament visual

### 2.4 Temps màxim d'alarma

Defineix el **temps màxim** que una alarma està activa si l'usuari no l'atura manualment.

Els temps màxims d'alarma disponibles per a aquesta versió són:

<b>10s</b>	Estableix un temps màxim d'alarma de 10s
<b>20s</b>	Estableix un temps màxim d'alarma de 20s
<b>30s</b>	Estableix un temps màxim d'alarma de 30s
<b>indefinit</b>	L'alarma està activa indefinidament

### 2.5 Volum de l'alarma

Defineix el percentatge de **volum de l'alarma** respecte el volum general configurat en el terminal.

Els percentatges de volum disponibles per a aquesta versió són:

<b>20%</b>	Estableix el volum al mínim
<b>40%</b>	Estableix el volum al 40%
<b>60%</b>	Estableix el volum al 60%
<b>80%</b>	Estableix el volum al 80%
<b>100%</b>	Estableix el volum al màxim



## Paràmetres de resum

3.1 **Codi PPStop** Indica el valor del **codi PPSTOP** que l'aplicació té configurat.

3.2 **Codi Historial** Indica el valor del **codi HISTORIAL** per a l'historial d'alarmes.

3.3 **Estat de Bluetooth** Indica l'estat de la connectivitat bluetooth del terminal mòbil.

**Actiu** Indica connectivitat activa  
**Inactiu** Indica connectivitat inactiva

3.4 **Estat de Servei** Indica l'estat actual d'associació al servei **PPSTOP**.

**No Associat** El terminal no està associat al servei PPSTOP  
**Associat** El terminal està associat al servei PPSTOP

3.5 **Ordre del terminal** Indica l'ordre del terminal dins de la memòria del dispositiu PPStop. El dispositiu PPStop generarà les alarmes seguint aquest ordre.

3.6 **Sensibilitat del sensor** Indica el nivell actual (baix-alt) de sensibilitat del sensor d'humitat del dispositiu **PPSTOP** associat.

**Baixa** Sensor amb sensibilitat mínima  
**Alta** Sensor amb sensibilitat màxima

3.7 **Darrera alarma** Indica el dia i l'hora en que s'ha rebut la darrera alarma. Addicionalment, també informa del temps que ha passat des de la darrera alarma.

3.8 **Pròxima acció** Indica quina és la propera acció que realitzarà l'aplicació quan rebí una connexió d'alarma del dispositiu **PPSTOP**. Aquesta acció es pot canviar a través de les opcions de **Buscar Terminals**, **Modificar Sensibilitat** i **Rotar Terminals**.

**Comprovar** Comprovar configuració i associar el servei (si escau)  
**Alarmar** Generar un avís d'alarma  
**Canvi sen.(X)** Canviar la sensibilitat del sensor d'humitat a X  
**Buscar term.** Iniciar una nova búsqueda de terminals PPSTOP  
**Rotar term.** Rotar cíclicament els terminals PPSTOP associats

# Alarma

## 4.1 Alarma

Quan l'aplicació rep una connexió d'**alarma** del dispositiu **PPSTOP**, s'activa l'alarma sonora (en cas que així estigui configurada l'aplicació) i també es mostra l'alarma de manera visual mitjançant l'aparició d'una campana.



Per atuar l'alarma cal pulsar l'opció **Aturar**.

Es pot comprovar la data i hora de la darrera alarma mitjançant el paràmetre **Darrera Alarma** de la **Pantalla de Resum**.

## 4.2 Historial d'alarmes

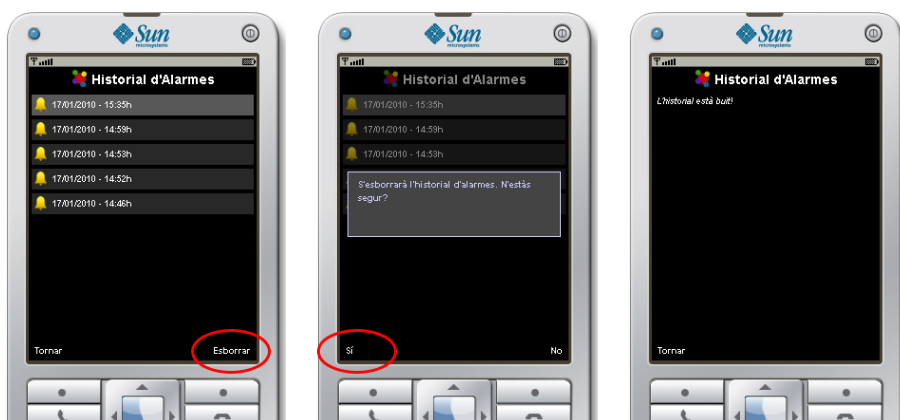
L'aplicació permet consultar l'**historial d'alarmes** enregistrades.

Per veure l'historial, cal accedir a l'opció **Historial d'Alarmes** del **Menú Principal**. Les alarmes apareixen llistades en ordre cronològicament invers: la més recent al principi i la més antiga al final.



## 4.3 Esborrar l'historial

L'**historial d'alarmes** es pot esborrar mitjançant l'opció **Esborrar** present a la part inferior-dreta de la pantalla de navegació de l'historial. Prèviament a l'esborrat, apareix un missatge de confirmació.





## Canvis sobre el dispositiu PPStop

### 5.1 Associar nous terminals al servei

Per a associar nous terminals al servei, s'ha d'indicar al dispositiu **PPSTOP** que procedeixi amb una nova búsqueda de terminals. Aquesta ordre es pot donar **NOMÉS** des d'un terminal ja associat al servei.

Per a fer-ho, s'han de seguir els següents passos:

- 1 Anar a l'opció "Buscar Terminals" del menú principal.
- 2 Acceptar el missatge de confirmació.
- 3 Apropar el terminal al dispositiu PPSTOP.
- 4 Activar voluntàriament el sensor d'humitat del dispositiu. El programa mostrarà un missatge emergent de "Buscant terminals...".
- 5 Esperar l'aparició d'un missatge al terminal indicant que el terminal s'ha associat novament al servei.



Cal tenir en compte que quan s'inicia una nova búsqueda de terminals, els terminals prèviament associats al servei **PPSTOP** deixen de ser-ho i cal que tornin a ser associats durant la nova búsqueda. Per tant, novament cal apropar al dispositiu **PPSTOP** tots els terminals que es vulguin associar.

### 5.2 Modificar la sensibilitat del sensor

El nivell de sensibilitat del sensor d'humitat del dispositiu **PPSTOP** es pot definir com a **alt** o **baix**.

**Baixa** Sensor amb sensibilitat mínima  
**Alta** Sensor amb sensibilitat màxima

Per a **canviar la sensibilitat** del sensor d'humitat del dispositiu s'han de seguir els següents passos:





- 1 Anar a l'opció "Modificar Sensibilitat" del menú principal.
- 2 Seleccionar la nova sensibilitat desitjada i acceptar.
- 3 Apropar el terminal al dispositiu PPSTOP.
- 4 Activar voluntàriament el sensor d'humitat del dispositiu.
- 5 Esperar l'aparició d'un missatge al terminal indicant que la modificació s'ha realitzat correctament.

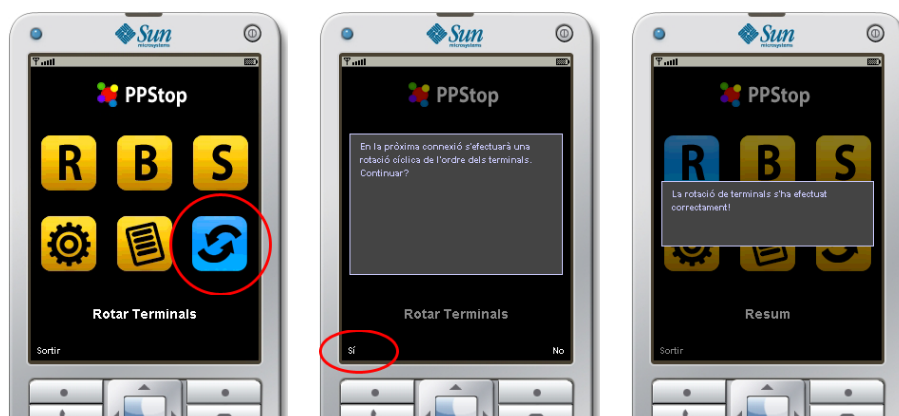


## 5.3 Rotar els terminals associats

Per a canviar l'ordre dels terminals associats al servei, s'ha d'indicar al dispositiu **PPSTOP** que procedeixi a rotar cíclicament l'ordre dels terminals. D'aquesta manera el primer terminal associat passarà a ser el segon, el segon a ser el tercer, i l'últim a ser el primer.

Per a fer-ho, s'han de seguir els següents passos:

- 1 Anar a l'opció "Rotar terminals" del menú principal.
- 2 Acceptar el missatge de confirmació.
- 3 Apropar el terminal al dispositiu PPSTOP.
- 4 Activar voluntàriament el sensor d'humitat del dispositiu.
- 5 Esperar l'aparició d'un missatge al terminal indicant que la rotació de terminals s'ha efectuat correctament.



## 5.4 Cancel·lar una acció sobre el dispositiu

Si s'ha activat la opció de **Buscar Terminals**, la de **Modificar Sensibilitat** o la de **Rotar Terminals**, aquesta es pot cancel·lar tornant a anar a una d'aquestes 3 opcions del **Menú Principal**. Cal respondre afirmativament a la pregunta de confirmació de cancel·lació.



Una vegada feta la cancel·lació, la pròxima acció a realitzar per l'aplicació davant d'una connexió del dispositiu **PPSTOP** torna a ser la d'**alarmar**. Es pot comprovar aquest paràmetre a través del camp **Pròxima Acció** de la **Pantalla de Resum**.



## Notes sobre l'aplicació

### 6.1 Requisits mínims

Els requisits mínims d'un terminal mòbil per al correcte funcionament de l'aplicació **PPSTOP** són els següents:

- Connectivitat Bluetooth
- Suport per a J2ME (MIDP 2.0 [JSR-118], CLDC 1.1 [JSR-139], JSR-82 i JSR-135 [JSR-185/JSR-248], KVM)
- 600KB d'espai lliure en memòria
- Resolució mínima de pantalla de 320px x 240px

### 6.2 Changelog

#### **Versió 3.1** (04/2010)

- Suport per a conèixer l'ordre del terminal
- Suport per a rotació cíclica dels terminals PPSTOP associats
- Suport per a actualització remota de paràmetres PPSTOP
- Reducció de la sensibilitat del sensor a 2 nivells

#### **Versió 3.0** (01/2010)

- Suport per a 3 idiomes
- Suport per a diferents tipus d'alarma
- Suport per a ajust del volum de l'alarma
- Suport per a temps màxim d'alarma
- Suport per a transferència d'història

#### **Versió 2.0** (12/2009)

- Nova interfície gràfica

#### **Versió 1.2** (07/2009)

- Correcció de bugs

#### **Versió 1.1** (06/2009)

- Suport per a història d'alarmes
- Suport per a búsqueda de terminals
- Suport per a modificació de sensibilitat

#### **Versió 1.0** (04/2009)

- Primera versió mínima funcional

### 6.3 Bugs i suggeriments

Aquesta versió de l'aplicació està en fase de proves. Es pot reportar qualsevol errada i/o deficiència que s'hi trobi, i també fer suggeriments per a millorar-la enviant un correu a la següent adreça de correu electrònic:

**Dr. Santiago Garcia-Tornel**  
**Oriol Ciurana**

stornel@hsjdbcn.es  
oriol.ciurana@entel.upc.edu

## C.5 Terminals mòbils compatibles

# Terminals compatibles amb PPStop

(\*) Només es llisten els terminals actualment en el mercat (01/2010)

Fabricant	Model
Nokia	X3
Nokia	X6
Nokia	2710 Navigation Edition
Nokia	2323 classic
Nokia	2330 classic
Nokia	2610
Nokia	2630
Nokia	2680 slide
Nokia	2690
Nokia	2700 classic
Nokia	2720 fold
Nokia	2730 classic
Nokia	2760
Nokia	3120 classic
Nokia	3600 slide
Nokia	3710 fold
Nokia	3720 classic
Nokia	5000
Nokia	5220 XpressMusic
Nokia	5230
Nokia	5310 XpressMusic
Nokia	5320 XpressMusic
Nokia	5330 XpressMusic
Nokia	5630 XpressMusic
Nokia	5730 XpressMusic
Nokia	5800 Navigation Edition
Nokia	5800 XpressMusic
Nokia	6124
Nokia	6210 Navigator
Nokia	6216 classic
Nokia	6220 classic
Nokia	6300
Nokia	6301
Nokia	6303 classic
Nokia	6500 classic
Nokia	6500 slide
Nokia	6555
Nokia	6600 fold
Nokia	6600i slide
Nokia	6700 classic
Nokia	6700 slide
Nokia	6710 Navigator

Fabricant	Model
Nokia	6720 classic
Nokia	6730 classic
Nokia	6760 slide
Nokia	7020
Nokia	7100 Supernova
Nokia	7210 Supernova
Nokia	7230
Nokia	7310 Supernova
Nokia	7610 Supernova
Nokia	8800 Carbon Arte
Nokia	N73
Nokia	N78
Nokia	N79
Nokia	N82
Nokia	N85
Nokia	N86 8MP
Nokia	N900
Nokia	N96
Nokia	N97
Nokia	N97 mini
Nokia	E52
Nokia	E55
Nokia	E63
Nokia	E65
Nokia	E66
Nokia	E71
Nokia	E72
Nokia	E75
Nokia	E90 Communicator
Sony Ericsson	W880i
Sony Ericsson	W580i
Sony Ericsson	K850i
Sony Ericsson	W910i
Sony Ericsson	K770i
Sony Ericsson	Z750
Sony Ericsson	W890i
Sony Ericsson	W380i
Sony Ericsson	T280i
Sony Ericsson	W760i
Sony Ericsson	C902
Sony Ericsson	C702
Sony Ericsson	G502

Fabricant	Model
Sony Ericsson	W980
Sony Ericsson	R300
Sony Ericsson	R306
Sony Ericsson	T303
Sony Ericsson	K330
Sony Ericsson	W350i
Sony Ericsson	W595
Sony Ericsson	Xperia Pureness
Sony Ericsson	W302
Sony Ericsson	T700
Sony Ericsson	J132
Sony Ericsson	Xperia X1
Sony Ericsson	W595s
Sony Ericsson	S302
Sony Ericsson	F305
Sony Ericsson	C905
Sony Ericsson	G705u
Sony Ericsson	G705
Sony Ericsson	C510
Sony Ericsson	W705
Sony Ericsson	W715
Sony Ericsson	W395
Sony Ericsson	W508
Sony Ericsson	W995
Sony Ericsson	S312
Sony Ericsson	T707
Sony Ericsson	C903
Sony Ericsson	C901 GreenHeart
Sony Ericsson	W205
Sony Ericsson	T715
Sony Ericsson	Jalou
Sony Ericsson	Naite
Sony Ericsson	Yari
Sony Ericsson	Aino
Sony Ericsson	Satio
Sony Ericsson	Xperia X2
Sony Ericsson	Xperia X10
Sony Ericsson	Vivaz
Samsung	B3410 Delphi
Samsung	C3212 Sling
Samsung	B7610 Omnia Pro
Samsung	S7350i
Samsung	E1360
Samsung	B2100
Samsung	C3050
Samsung	C3060
Samsung	C3110
Samsung	C6620

Fabricant	Model
Samsung	C6625
Samsung	E1310M
Samsung	E2100
Samsung	E2210B
Samsung	E2510
Samsung	I8000 Omnia 2
Samsung	i8510 Innov8
Samsung	i8910
Samsung	M3200
Samsung	M3510
Samsung	M7600 Beat Dj
Samsung	M8800 Pixon
Samsung	M8910 Pixon 12
Samsung	S3030
Samsung	S5050
Samsung	S5230 Star
Samsung	S5600 Hy Touch
Samsung	S6700
Samsung	S7330
Samsung	S8000 Jet
Samsung	S8300 UltraTouch
Samsung	B3310
Samsung	B7300 Omnia Lite
Samsung	I7500 Galaxy
Samsung	M2510
Samsung	S3650
Samsung	S5200
Samsung	C180
Samsung	D880 DOSS
Samsung	D980
Samsung	E200 Eco
Samsung	E250
Samsung	E840
Samsung	F110 MiCoach
Samsung	F210
Samsung	F330
Samsung	F400
Samsung	F480
Samsung	F700
Samsung	G600
Samsung	G800
Samsung	G810
Samsung	i200
Samsung	i450
Samsung	i560
Samsung	i780
Samsung	I900 Omnia
Samsung	J400

Fabricant	Model
Samsung	J700i
Samsung	J700
Samsung	J800
Samsung	L170
Samsung	L700
Samsung	L760
Samsung	L770
Samsung	L870
Samsung	M310
Samsung	P520
Samsung	U100
Samsung	U600
Samsung	U700
Samsung	U900 Soul
Samsung	Z170
Samsung	Z240
Motorola	DEXT
Motorola	AURA
Motorola	MOTOJEWEL
Motorola	MOTOKRZR K1
Motorola	MOTOKRZR K3
Motorola	MOTOQ 9h
Motorola	MOTORAZR2 V8
Motorola	MOTORAZR2
Motorola	MOTORAZR V3
Motorola	MOTORAZR V3i
Motorola	MOTORIZR Z3
Motorola	MOTOROKR Z6
Motorola	MOTOROKR Z6w
Motorola	MOTOSLVR L6
Motorola	MOTOSLVR L9
Motorola	MOTOZINE ZN5
Motorola	MOTO EM325
Motorola	MOTO U9
Motorola	MOTO VE538
Motorola	MOTO VE66
Motorola	ROKR E8
Motorola	W377
Motorola	W510
LG	BL40
LG	KP500
LG	KC910
LG	KF310
LG	U970
LG	KG800
LG	KP235
LG	KF750
LG	HB620T

Fabricant	Model
LG	KU990
LG	KF700
LG	KF510



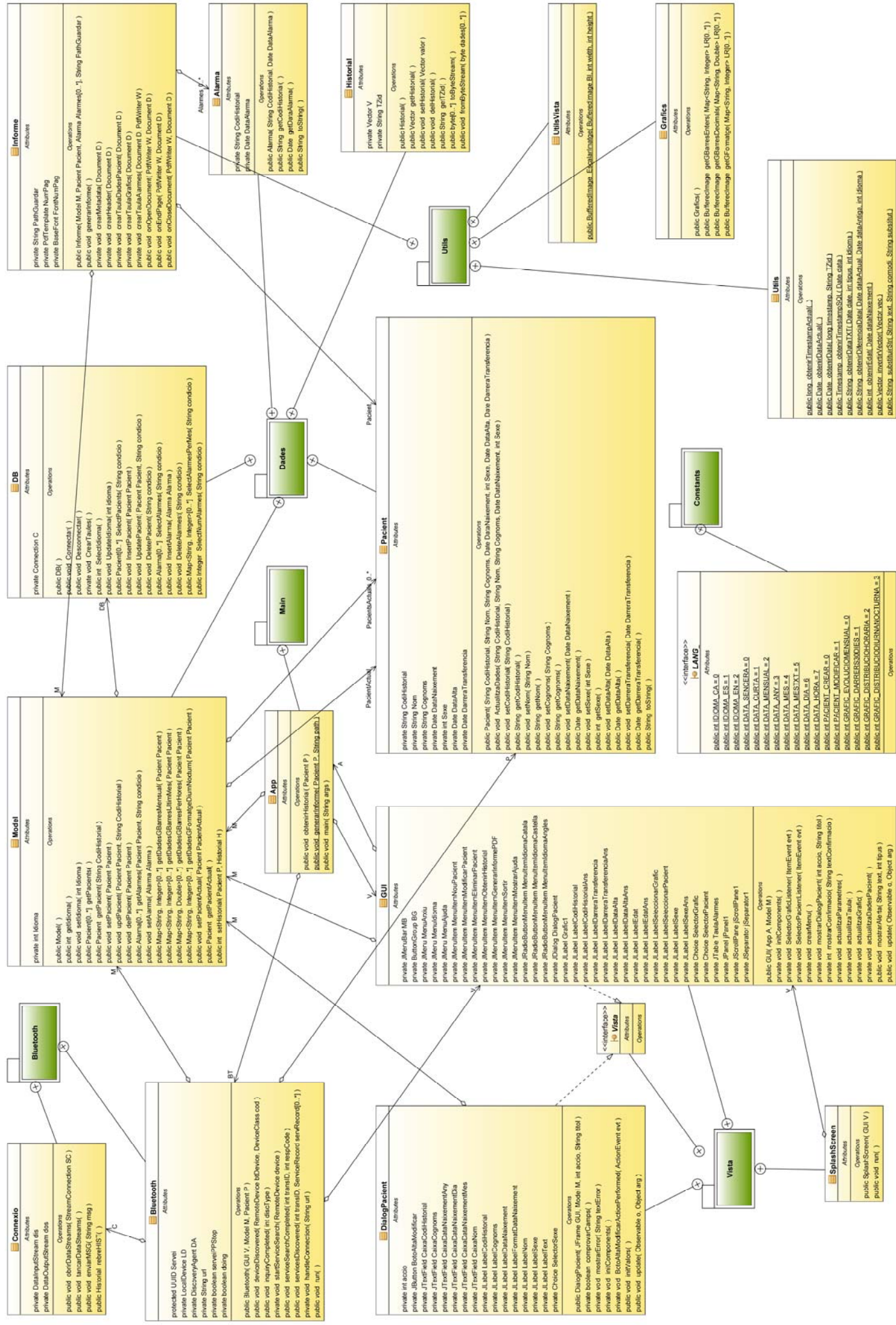


## Apèndix D

### Aplicació de seguiment



## D.1 Diagrama de classes de l'aplicació



## D.2 JavaDoc de l'aplicació

---

# Package Bluetooth

Conté les classes per a realitzar una connexió bluetooth i gestionar tota comunicació amb l'aplicació

## Bluetooth

### Class Bluetooth

```

java.lang.Object
  |
  +- java.lang.Thread
        |
        +- Bluetooth.Bluetooth
  
```

#### All Implemented Interfaces:

javax.bluetooth.DiscoveryListener, java.lang.Runnable

```

public class Bluetooth
extends java.lang.Thread
implements java.lang.Runnable, javax.bluetooth.DiscoveryListener
  
```

Classe que obté l'historial d'alarmes d'un pacient mitjançant Bluetooth

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

1.0 (Juny de 2010)

Field Summary		
private	<a href="#"><u>C</u></a>	Objecte de connexió
private	<a href="#"><u>DA</u></a>	Agent de descobriment
private	<a href="#"><u>doing</u></a>	Identifica el final de la transferència d'historial
private	<a href="#"><u>LD</u></a>	Dispositiu local
private	<a href="#"><u>M</u></a>	Referència al Model
private	<a href="#"><u>P</u></a>	Pacient del que es vol obtenir l'Historial
protected	<a href="#"><u>Servei</u></a>	UUID del servei a buscar mitjançant el SDP
private	<a href="#"><u>serveiPPStop</u></a>	Indica si s'ha trobat un servei PPStop o no
private	<a href="#"><u>url</u></a>	URL de connexió
private	<a href="#"><u>V</u></a>	Referència a la GUI on mostrar les alertes

Fields inherited from class `java.lang.Thread`

blocker, blockerLock, contextClassLoader, daemon, defaultUncaughtExceptionHandler, eetop, EMPTY\_STACK\_TRACE, group, inheritableThreadLocals, inheritedAccessControlContext, MAX\_PRIORITY, MIN\_PRIORITY, name, nativeParkEventPointer, NORM\_PRIORITY, parkBlocker, priority, single\_step, stackSize, stillborn, stopBeforeStart, SUBCLASS\_IMPLEMENTATION\_PERMISSION, subclassAudits, target, threadInitNumber, threadLocals, threadQ, threadSeqNumber, threadStatus, throwableFromStop, tid, uncaughtExceptionHandler

#### Fields inherited from interface `javax.bluetooth.DiscoveryListener`

INQUIRY\_COMPLETED, INQUIRY\_ERROR, INQUIRY\_TERMINATED, SERVICE\_SEARCH\_COMPLETED, SERVICE\_SEARCH\_DEVICE\_NOT\_REACHABLE, SERVICE\_SEARCH\_ERROR, SERVICE\_SEARCH\_NO\_RECORDS, SERVICE\_SEARCH\_TERMINATED

## Constructor Summary

public	<a href="#">Bluetooth</a> ( <a href="#">GUI</a> V, <a href="#">Model</a> M, <a href="#">Pacient</a> P) Crea una instància de Bluetooth
--------	---

## Method Summary

void	<a href="#">deviceDiscovered</a> ( <code>javax.bluetooth.RemoteDevice</code> btDevice, <code>javax.bluetooth.DeviceClass</code> cod) Mètode que és cridat quan s'ha descobert un nou dispositiu a través del SDP
void	<a href="#">handleConnection</a> ( <code>java.lang.String</code> url) Realitza la connexió amb el dispositiu remot i n'obté l'historial d'alarmes
void	<a href="#">inquiryCompleted</a> ( <code>int</code> discType) Mètode que és cridat quan s'ha completat la búsqueda de dispositius (DDP)
void	<a href="#">run</a> ()
void	<a href="#">servicesDiscovered</a> ( <code>int</code> transID, <code>javax.bluetooth.ServiceRecord[]</code> servRecord) Mètode que és cridat quan s'ha completat la búsqueda de serveis (SDP)
void	<a href="#">serviceSearchCompleted</a> ( <code>int</code> transID, <code>int</code> respCode) Mètode que és cridat quan s'ha completat la búsqueda de serveis (SDP)
void	<a href="#">startServiceSearch</a> ( <code>javax.bluetooth.RemoteDevice</code> device) Inicia una búsqueda de serveis (SDP) en el dispositiu especificat

#### Methods inherited from class `java.lang.Thread`

activeCount, auditSubclass, blockedOn, checkAccess, countStackFrames, currentThread, destroy, dispatchUncaughtException, dumpStack, dumpThreads, enumerate, exit, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getThreads, getUncaughtExceptionHandler, holdsLock, init, interrupt, interrupt0, interrupted, isAlive, isCCLOverridden, isDaemon, isInterrupted, isInterrupted, join, join, join, nextThreadID, nextThreadNum, registerNatives, resume, resume0, run, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setPriority0, setUncaughtExceptionHandler, sleep, sleep, start, start0, stop, stop, stop0, stop1, suspend, suspend0, toString, yield

#### Methods inherited from class `java.lang.Object`



```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

**Methods inherited from interface** `java.lang.Runnable`

```
run
```

**Methods inherited from interface** `javax.bluetooth.DiscoveryListener`

```
deviceDiscovered, inquiryCompleted, servicesDiscovered, serviceSearchCompleted
```

## Fields

### V

```
private Vista.GUI V
```

Referència a la GUI on mostrar les alertes

### M

```
private Dades.Model M
```

Referència al Model

### P

```
private Dades.Pacient P
```

Pacient del que es vol obtenir l'Historial

### Servei

```
protected javax.bluetooth.UUID Servei
```

UUID del servei a buscar mitjançant el SDP

### C

```
private Bluetooth.Connexio C
```

Objecte de connexió

### LD

```
private javax.bluetooth.LocalDevice LD
```

Dispositiu local

### DA

```
private javax.bluetooth.DiscoveryAgent DA
```

Agent de descobriment

## url

```
private java.lang.String url
```

URL de connexió

---

## serveiPPStop

```
private boolean serveiPPStop
```

Indica si s'ha trobat un servei PPStop o no

---

## doing

```
private boolean doing
```

Identifica el final de la transferència d'història

---

## Constructors

### Bluetooth

```
public Bluetooth(GUI V,  
                 Model M,  
                 Pacient P)
```

Crea una instància de Bluetooth

#### Parameters:

M - GUI on mostrar les alertes corresponents

M - Model de dades

P - Pacient del que es vol obtenir l'història

## Methods

### deviceDiscovered

```
public void deviceDiscovered(javax.bluetooth.RemoteDevice btDevice,  
                             javax.bluetooth.DeviceClass cod)
```

Mètode que és cridat quan s'ha descobert un nou dispositiu a través del SDP

#### Parameters:

btDevice - Dispositiu remot descobert

cod - Codi identificador de la classe de dispositiu descobert

---

### inquiryCompleted

```
public void inquiryCompleted(int discType)
```

Mètode que és cridat quan s'ha completat la búsqueda de dispositius (DDP)

#### Parameters:

discType - Identificador del motiu de la finalització DDP

---

(continued from last page)

## startServiceSearch

```
private void startServiceSearch(javax.bluetooth.RemoteDevice device)
```

Inicia una búsqueda de serveis (SDP) en el dispositiu especificat

**Parameters:**

device - Dispositiu sobre el que iniciar el SDP

---

## serviceSearchCompleted

```
public void serviceSearchCompleted(int transID,  
int respCode)
```

Mètode que és cridat quan s'ha completat la búsqueda de serveis (SDP)

**Parameters:**

transID - Identificador del recurs que ha iniciat el SDP

respCode - Codi identificador de l'estat de la transacció

---

## servicesDiscovered

```
public void servicesDiscovered(int transID,  
javax.bluetooth.ServiceRecord[] servRecord)
```

Mètode que és cridat quan s'ha completat la búsqueda de serveis (SDP)

**Parameters:**

transID - Identificador del recurs que ha apuntat el resultat

respCode - Llista de seveis trobats durant la búsqueda de serveis (SDP)

---

## handleConnection

```
private void handleConnection(java.lang.String url)
```

Realitza la connexió amb el dispositiu remot i n'obté l'historial d'alarmes

**Parameters:**

url - Adreça de connexió

---

## run

```
public void run()
```

# Bluetooth

## Class Connexio

```
java.lang.Object
|
+-Bluetooth.Connexio
```

```
public class Connexio
extends java.lang.Object
```

Classe que obre els streams d'e I/O per a la transmissió i recepció de dades

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
1.0 (Juny de 2010)

### Field Summary

private	<a href="#"><u>dis</u></a> Stream d'entrada de dades
private	<a href="#"><u>dos</u></a> Stream de sortida de dades

### Constructor Summary

public	<a href="#"><u>Connexio()</u></a>
--------	-----------------------------------

### Method Summary

void	<a href="#"><u>enviarMSG()</u></a> (java.lang.String msg) Transmet un Missatge a través de l'OutputStream
void	<a href="#"><u>obrirDataStreams()</u></a> (javax.microedition.io.StreamConnection SC) Obtené els Streams de I/O a partir d'un l'StreamConnection
<a href="#"><u>Historial</u></a>	<a href="#"><u>rebreHIST()</u></a> Rep un Historial a través de l'OutputStream
void	<a href="#"><u>tancarDataStreams()</u></a> Tanca els Streams de I/O

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### dis

```
private java.io.DataInputStream dis
```

Stream d'entrada de dades

---

## **dos**

```
private java.io.DataOutputStream dos
```

Stream de sortida de dades

## Constructors

### **Connexio**

```
public Connexio()
```

## Methods

### **obrirDataStreams**

```
public void obrirDataStreams(javax.microedition.io.StreamConnection SC)
```

Obtené els Streams de I/O a partir d'un l'StreamConnection

**Parameters:**

SC - Stream de la connexió

---

### **tancarDataStreams**

```
public void tancarDataStreams()
```

Tanca els Streams de I/O

---

### **enviarMSG**

```
public void enviarMSG(java.lang.String msg)
```

Transmet un Missatge a través de l'OutputStream

**Parameters:**

msg - Missatge a enviar

---

### **rebreHIST**

```
public Historial rebreHIST()
```

Rep un Historial a través de l'OutputStream

**Returns:**

Historial rebut

---

# Package Constants

## Constants

### Interface CONST

public interface **CONST**  
extends

Interfície que conté totes les constants numèriques utilitzades per l'aplicació

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juny de 2010)

#### Field Summary

public static final	<a href="#"><u>DATA_ANY</u></a> El tipus de data és l'any Value: 3
public static final	<a href="#"><u>DATA_CURTA</u></a> El tipus de data és la data curta Value: 1
public static final	<a href="#"><u>DATA_DIA</u></a> El tipus de data és el dia Value: 6
public static final	<a href="#"><u>DATA_HORA</u></a> El tipus de data és l'hora Value: 7
public static final	<a href="#"><u>DATA_MENSUAL</u></a> El tipus de data és la data mensual Value: 2
public static final	<a href="#"><u>DATA_MES</u></a> El tipus de data és el mes Value: 4
public static final	<a href="#"><u>DATA_MESTXT</u></a> El tipus de data és el mes en format text Value: 5
public static final	<a href="#"><u>DATA_SENCERA</u></a> El tipus de data és la data sencera Value: 0
public static final	<a href="#"><u>GRAFIC_DARRERS30DIES</u></a> El tipus de gràfic a mostrar és el de l'evolució d'alarmes dels darrers 30 dies Value: 1
public static final	<a href="#"><u>GRAFIC_DISTRIBUCIODIURNANOCTURNA</u></a> El tipus de gràfic a mostrar és el de la distribució diurna/nocturna de les alarmes Value: 3

public static final	<a href="#">GRAFIC_DISTRIBUCIOHORARIA</a> El tipus de gràfic a mostrar és el de la distribució horària de les alarmes Value: 2
public static final	<a href="#">GRAFIC_EVOLUCIOMENSUAL</a> El tipus de gràfic a mostrar és el de l'evolució d'alarmes mensual Value: 0
public static final	<a href="#">IDIOMA_CA</a> L'idioma de la interfície de l'aplicació és el Català Value: 0
public static final	<a href="#">IDIOMA_EN</a> L'idioma de la interfície de l'aplicació és l'Anglès Value: 2
public static final	<a href="#">IDIOMA_ES</a> L'idioma de la interfície de l'aplicació és el Castellà Value: 1
public static final	<a href="#">PACIENT_CREAR</a> El tipus d'acció és la de crear un nou pacient Value: 0
public static final	<a href="#">PACIENT_MODIFICAR</a> El tipus d'acció és la de modificar el pacient que s'està tractant Value: 1

## Fields

### IDIOMA\_CA

public static final int **IDIOMA\_CA**

L'idioma de la interfície de l'aplicació és el Català  
Constant value: 0

### IDIOMA\_ES

public static final int **IDIOMA\_ES**

L'idioma de la interfície de l'aplicació és el Castellà  
Constant value: 1

### IDIOMA\_EN

public static final int **IDIOMA\_EN**

L'idioma de la interfície de l'aplicació és l'Anglès  
Constant value: 2

### DATA\_SENCERA

public static final int **DATA\_SENCERA**

El tipus de data és la data sencera



(continued from last page)

Constant value: 0

---

## DATA\_CURTA

```
public static final int DATA_CURTA
```

El tipus de data és la data curta  
Constant value: 1

---

## DATA\_MENSUAL

```
public static final int DATA_MENSUAL
```

El tipus de data és la data mensual  
Constant value: 2

---

## DATA\_ANY

```
public static final int DATA_ANY
```

El tipus de data és l'any  
Constant value: 3

---

## DATA\_MES

```
public static final int DATA_MES
```

El tipus de data és el mes  
Constant value: 4

---

## DATA\_MESTXT

```
public static final int DATA_MESTXT
```

El tipus de data és el mes en format text  
Constant value: 5

---

## DATA\_DIA

```
public static final int DATA_DIA
```

El tipus de data és el dia  
Constant value: 6

---

## DATA\_HORA

```
public static final int DATA_HORA
```

El tipus de data és l'hora  
Constant value: 7

---

## PACIENT\_CREAR

```
public static final int PACIENT_CREAR
```

El tipus d'acció és la de crear un nou pacient  
Constant value: 0

---

(continued from last page)

## PACIENT\_MODIFICAR

```
public static final int PACIENT_MODIFICAR
```

El tipus d'acció és la de modificar el pacient que s'està tractant  
Constant value: **1**

---

## GRAFIC\_EVOLUCIOMENSUAL

```
public static final int GRAFIC_EVOLUCIOMENSUAL
```

El tipus de gràfic a mostrar és el de l'evolució d'alarmes mensual  
Constant value: **0**

---

## GRAFIC\_DARRERS30DIES

```
public static final int GRAFIC_DARRERS30DIES
```

El tipus de gràfic a mostrar és el de l'evolució d'alarmes dels darrers 30 dies  
Constant value: **1**

---

## GRAFIC\_DISTRIBUCIOHORARIA

```
public static final int GRAFIC_DISTRIBUCIOHORARIA
```

El tipus de gràfic a mostrar és el de la distribució horària de les alarmes  
Constant value: **2**

---

## GRAFIC\_DISTRIBUCIODIURNANOCTURNA

```
public static final int GRAFIC_DISTRIBUCIODIURNANOCTURNA
```

El tipus de gràfic a mostrar és el de la distribució diurna/nocturna de les alarmes  
Constant value: **3**

---

## Constants

### Interface LANG

public interface **LANG**  
extends

Interfície que conté totes les constants d'idioma utilitzades per l'aplicació

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juny de 2010)

#### Field Summary

public static final	<a href="#">MENU_AJUDA</a> Label del menú Ajuda
public static final	<a href="#">MENU_ANGLES</a> Label del menú Idioma: Anglès
public static final	<a href="#">MENU_ARXIU</a> Label del menú Arxiu
public static final	<a href="#">MENU_CASTELLA</a> Label del menú Idioma: Castellà
public static final	<a href="#">MENU_CATALA</a> Label del menú Idioma: Català
public static final	<a href="#">MENU_ELIMINARPACIENT</a> Label del menú Arxiu: Modificar Pacient
public static final	<a href="#">MENU_GENERARINFORMEPDF</a> Label del menú Arxiu: Generar Informe PDF
public static final	<a href="#">MENU_IDIOMA</a> Label del menú Idioma
public static final	<a href="#">MENU_MODIFICARPACIENT</a> Label del menú Arxiu: Modificar Pacient
public static final	<a href="#">MENU_MOSTRARAJUDA</a> Label del menú Ajuda: Mostrar Ajuda
public static final	<a href="#">MENU_NOUPACIENT</a> Label del menú Arxiu: Nou Pacient
public static final	<a href="#">MENU_OBTENIRHISTORIAL</a> Label del menú Arxiu: Obtenir Historial
public static final	<a href="#">MENU_SORTIR</a> Label del menú Arxiu: Sortir
public static final	<a href="#">TXT_1DIA</a> Label de diferència de temps: dia

public static final	<a href="#"><u>TXT_1HORA</u></a> Label de diferència de temps: 1 hora
public static final	<a href="#"><u>TXT_1MINUT</u></a> Label de diferència de temps: minut
public static final	<a href="#"><u>TXT_ALERTA_HISTORIALREBUT</u></a> Label d'alerta: historial rebut
public static final	<a href="#"><u>TXT_ALERTA_OPERACIOPOTTRIGAR</u></a> Label d'alerta: l'operació pot trigar uns segons
public static final	<a href="#"><u>TXT_ANY</u></a> Label d'any
public static final	<a href="#"><u>TXT_ANYS</u></a> Label d'anys
public static final	<a href="#"><u>TXT_CODIHISTORIAL</u></a> Label de codi d'historial
public static final	<a href="#"><u>TXT_COGNOMS</u></a> Label de cognoms
public static final	<a href="#"><u>TXT_CONFIRMACIO_ELIMINAR</u></a> Label de confirmació: eliminar pacient
public static final	<a href="#"><u>TXT_DADESPACIENT</u></a> Label de dades del pacient
public static final	<a href="#"><u>TXT_DARRERATRANSFERENCIA</u></a> Label de darrera transferència
public static final	<a href="#"><u>TXT_DATAALTA</u></a> Label de data d'alta
public static final	<a href="#"><u>TXT_DATAGENERACIO</u></a> Label de data de generació
public static final	<a href="#"><u>TXT_DATANAIXEMENT</u></a> Label de data de naixement
public static final	<a href="#"><u>TXT_DIA</u></a> Label de dia
public static final	<a href="#"><u>TXT_DIALOGPACIENT1</u></a> Labrl d'introducció de dades del nou pacient
public static final	<a href="#"><u>TXT_DIALOGPACIENT2</u></a> Label de modificació de dades d'un pacient
public static final	<a href="#"><u>TXT_DIURNES</u></a> Label d'alarmes diürnes
public static final	<a href="#"><u>TXT_DONARDALTA</u></a> Label d'alta de pacient
public static final	<a href="#"><u>TXT_EDAT</u></a> Label d'edat

public static final	<a href="#"><u>TXT_ERROR_CODIHISTORIAL</u></a> Label d'error: codi d'historial no vàlid
public static final	<a href="#"><u>TXT_ERROR_CODIHISTORIALEXISTEIX</u></a> Label d'error: codi d'historial ja existent
public static final	<a href="#"><u>TXT_ERROR_COGNOMS</u></a> Label d'error: cognoms buit
public static final	<a href="#"><u>TXT_ERROR_DATANAIXEMENT</u></a> Label d'error: data de naixement no vàlida
public static final	<a href="#"><u>TXT_ERROR_GENERACIOINFORME</u></a> Label d'error: no es pot generar l'informe perquè no hi ha alarmes
public static final	<a href="#"><u>TXT_ERROR_NOALARMES</u></a> Label d'error: no es pot generar l'informe perquè no hi ha alarmes
public static final	<a href="#"><u>TXT_ERROR_NOBLUETOOTH</u></a> Label d'error: no hi ha bluetooth
public static final	<a href="#"><u>TXT_ERROR_NOM</u></a> Label d'error: nom buit
public static final	<a href="#"><u>TXT_ERROR_NOPACIENT</u></a> Label d'error: no hi ha cap pacient seleccionat
public static final	<a href="#"><u>TXT_ERROR_NOSERVEIPPSTOP</u></a> Label d'error: no s'ha trobat servei PPStop
public static final	<a href="#"><u>TXT_ERROR_TRANSFERENCEIAHISTORIAL</u></a> Label d'error: error en la transferència d'historial
public static final	<a href="#"><u>TXT_GRAFIC1</u></a> Label de la gràfica de tipus 1
public static final	<a href="#"><u>TXT_GRAFIC2</u></a> Label de la gràfica de tipus 2
public static final	<a href="#"><u>TXT_GRAFIC3</u></a> Label de la gràfica de tipus 3
public static final	<a href="#"><u>TXT_GRAFIC4</u></a> Label de la gràfica de tipus 4
public static final	<a href="#"><u>TXT_GRAFIQUESESTADISTIQUES</u></a> Label de grafiques estadístiques
public static final	<a href="#"><u>TXT_HISTORIALALARMES</u></a> Label d'historial d'alarmes
public static final	<a href="#"><u>TXT_HORA</u></a> Label d'hora
public static final	<a href="#"><u>TXT_INFORME</u></a> Label del títol de l'informe de seguiment
public static final	<a href="#"><u>TXT_MENYSDUNMINUT</u></a> Label de diferència de temps: < 1 minut

<code>public static final</code>	<a href="#"><u>TXT_MES</u></a> Label de mes
<code>public static final</code>	<a href="#"><u>TXT_MESTXT</u></a> Label de tipus de mes
<code>public static final</code>	<a href="#"><u>TXT_MODIFICARDADES</u></a> Label de modificacio de pacient
<code>public static final</code>	<a href="#"><u>TXT_NOCTURNES</u></a> Label d'alarmes nocturnes
<code>public static final</code>	<a href="#"><u>TXT_NOM</u></a> Label de nom
<code>public static final</code>	<a href="#"><u>TXT_PAGINADE</u></a> Label d'indicador de pàgina
<code>public static final</code>	<a href="#"><u>TXT_SELECCIONARGRAFIC</u></a> Label de selecció de gràfic
<code>public static final</code>	<a href="#"><u>TXT_SELECCIONARPACIENT</u></a> Label de selecció de pacient
<code>public static final</code>	<a href="#"><u>TXT_SEXE</u></a> Label de sexe
<code>public static final</code>	<a href="#"><u>TXT_SEXETXT</u></a> Label de tipus de sexe
<code>public static final</code>	<a href="#"><u>TXT_XDIES</u></a> Label de diferència de temps: dies
<code>public static final</code>	<a href="#"><u>TXT_XHORES</u></a> Label de diferència de temps: hores
<code>public static final</code>	<a href="#"><u>TXT_XMINUTS</u></a> Label de diferència de temps: minuts

## Fields

### TXT\_INFORME

```
public static final java.lang.String TXT_INFORME
```

Label del títol de l'informe de seguiment

### TXT\_DATAGENERACIO

```
public static final java.lang.String TXT_DATAGENERACIO
```

Label de data de generació

### TXT\_DADESPACIENT

```
public static final java.lang.String TXT_DADESPACIENT
```

(continued from last page)

Label de dades del pacient

---

## **TXT\_CODIHISTORIAL**

```
public static final java.lang.String TXT_CODIHISTORIAL
```

Label de codi d'historial

---

## **TXT\_NOM**

```
public static final java.lang.String TXT_NOM
```

Label de nom

---

## **TXT\_COGNOMS**

```
public static final java.lang.String TXT_COGNOMS
```

Label de cognoms

---

## **TXT\_EDAT**

```
public static final java.lang.String TXT_EDAT
```

Label d'edat

---

## **TXT\_DATANAIXEMENT**

```
public static final java.lang.String TXT_DATANAIXEMENT
```

Label de data de naixement

---

## **TXT\_SEXE**

```
public static final java.lang.String TXT_SEXE
```

Label de sexe

---

## **TXT\_SEXETXT**

```
public static final java.lang.String TXT_SEXETXT
```

Label de tipus de sexe

---

## **TXT\_DATAALTA**

```
public static final java.lang.String TXT_DATAALTA
```

Label de data d'alta

---

## **TXT\_DARRERATRANSFERENCIA**

```
public static final java.lang.String TXT_DARRERATRANSFERENCIA
```

Label de darrera transferència

(continued from last page)

## **TXT\_GRAFIQUESESTADISTIQUES**

```
public static final java.lang.String TXT_GRAFIQUESESTADISTIQUES
```

Label de grafiques estadísticas

---

## **TXT\_HISTORIALALARMES**

```
public static final java.lang.String TXT_HISTORIALALARMES
```

Label d'historial d'alarmes

---

## **TXT\_ANY**

```
public static final java.lang.String TXT_ANY
```

Label d'any

---

## **TXT\_ANYS**

```
public static final java.lang.String TXT_ANYS
```

Label d'anys

---

## **TXT\_MES**

```
public static final java.lang.String TXT_MES
```

Label de mes

---

## **TXT\_MESTXT**

```
public static final java.lang.String TXT_MESTXT
```

Label de tipus de mes

---

## **TXT\_DIA**

```
public static final java.lang.String TXT_DIA
```

Label de dia

---

## **TXT\_HORA**

```
public static final java.lang.String TXT_HORA
```

Label d'hora

---

## **TXT\_DIURNES**

```
public static final java.lang.String TXT_DIURNES
```

Label d'alarmes diürnes

---

## **TXT\_NOCTURNES**

```
public static final java.lang.String TXT_NOCTURNES
```



(continued from last page)

Label d'alarmes nocturnes

---

## **TXT\_PAGINADE**

```
public static final java.lang.String TXT_PAGINADE
```

Label d'indicador de pàgina

---

## **TXT\_GRAFIC1**

```
public static final java.lang.String TXT_GRAFIC1
```

Label de la gràfica de tipus 1

---

## **TXT\_GRAFIC2**

```
public static final java.lang.String TXT_GRAFIC2
```

Label de la gràfica de tipus 2

---

## **TXT\_GRAFIC3**

```
public static final java.lang.String TXT_GRAFIC3
```

Label de la gràfica de tipus 3

---

## **TXT\_GRAFIC4**

```
public static final java.lang.String TXT_GRAFIC4
```

Label de la gràfica de tipus 4

---

## **TXT\_MENYSDUNMINUT**

```
public static final java.lang.String TXT_MENYSDUNMINUT
```

Label de diferència de temps: < 1 minut

---

## **TXT\_1MINUT**

```
public static final java.lang.String TXT_1MINUT
```

Label de diferència de temps: minut

---

## **TXT\_XMINUTS**

```
public static final java.lang.String TXT_XMINUTS
```

Label de diferència de temps: minuts

---

## **TXT\_1HORA**

```
public static final java.lang.String TXT_1HORA
```

Label de diferència de temps: 1 hora

(continued from last page)

## **TXT\_XHORES**

```
public static final java.lang.String TXT_XHORES
```

Label de diferència de temps: hores

---

## **TXT\_1DIA**

```
public static final java.lang.String TXT_1DIA
```

Label de diferència de temps: dia

---

## **TXT\_XDIES**

```
public static final java.lang.String TXT_XDIES
```

Label de diferència de temps: dies

---

## **MENU\_ARXIU**

```
public static final java.lang.String MENU_ARXIU
```

Label del menú Arxiu

---

## **MENU\_IDIOMA**

```
public static final java.lang.String MENU_IDIOMA
```

Label del menú Idioma

---

## **MENU\_AJUDA**

```
public static final java.lang.String MENU_AJUDA
```

Label del menú Ajuda

---

## **MENU\_NOUPACIENT**

```
public static final java.lang.String MENU_NOUPACIENT
```

Label del menú Arxiu: Nou Pacient

---

## **MENU\_MODIFICARPACIENT**

```
public static final java.lang.String MENU_MODIFICARPACIENT
```

Label del menú Arxiu: Modificar Pacient

---

## **MENU\_ELIMINARPACIENT**

```
public static final java.lang.String MENU_ELIMINARPACIENT
```

Label del menú Arxiu: Modificar Pacient

---

## **MENU\_OBTENIRHISTORIAL**

```
public static final java.lang.String MENU_OBTENIRHISTORIAL
```

(continued from last page)

Label del menú Arxiu: Obtenir Historial

---

## MENU\_GENERARINFORMEPDF

```
public static final java.lang.String MENU_GENERARINFORMEPDF
```

Label del menú Arxiu: Generar Informe PDF

---

## MENU\_SORTIR

```
public static final java.lang.String MENU_SORTIR
```

Label del menú Arxiu: Sortir

---

## MENU\_CATALA

```
public static final java.lang.String MENU_CATALA
```

Label del menú Idioma: Català

---

## MENU\_CASTELLA

```
public static final java.lang.String MENU_CASTELLA
```

Label del menú Idioma: Castellà

---

## MENU\_ANGLES

```
public static final java.lang.String MENU_ANGLES
```

Label del menú Idioma: Anglès

---

## MENU\_MOSTRARAJUDA

```
public static final java.lang.String MENU_MOSTRARAJUDA
```

Label del menú Ajuda: Mostrar Ajuda

---

## TXT\_SELECCIONARPACIENT

```
public static final java.lang.String TXT_SELECCIONARPACIENT
```

Label de selecció de pacient

---

## TXT\_SELECCIONARGRAFIC

```
public static final java.lang.String TXT_SELECCIONARGRAFIC
```

Label de selecció de gràfic

---

## TXT\_DIALOGPACIENT1

```
public static final java.lang.String TXT_DIALOGPACIENT1
```

Labrl d'introducció de dades del nou pacient

(continued from last page)

---

## **TXT\_DIALOGPACIENT2**

```
public static final java.lang.String TXT_DIALOGPACIENT2
```

Label de modificació de dades d'un pacient

---

## **TXT\_DONARDALTA**

```
public static final java.lang.String TXT_DONARDALTA
```

Label d'alta de pacient

---

## **TXT\_MODIFICARDADES**

```
public static final java.lang.String TXT_MODIFICARDADES
```

Label de modificacio de pacient

---

## **TXT\_CONFIRMACIO\_ELIMINAR**

```
public static final java.lang.String TXT_CONFIRMACIO_ELIMINAR
```

Label de confirmació: eliminar pacient

---

## **TXT\_ERROR\_NOM**

```
public static final java.lang.String TXT_ERROR_NOM
```

Label d'error: nom buit

---

## **TXT\_ERROR\_COGNOMS**

```
public static final java.lang.String TXT_ERROR_COGNOMS
```

Label d'error: cognoms buit

---

## **TXT\_ERROR\_DATANAIXEMENT**

```
public static final java.lang.String TXT_ERROR_DATANAIXEMENT
```

Label d'error: data de naixement no vàlida

---

## **TXT\_ERROR\_CODIHISTORIAL**

```
public static final java.lang.String TXT_ERROR_CODIHISTORIAL
```

Label d'error: codi d'historial no vàlid

---

## **TXT\_ERROR\_CODIHISTORIALEXISTEIX**

```
public static final java.lang.String TXT_ERROR_CODIHISTORIALEXISTEIX
```

Label d'error: codi d'historial ja existent

---

## **TXT\_ERROR\_NOALARMES**

```
public static final java.lang.String TXT_ERROR_NOALARMES
```

---

(continued from last page)

Label d'error: no es pot generar l'informe perquè no hi ha alarmes

---

## **TXT\_ERROR\_GENERACIOINFORME**

```
public static final java.lang.String TXT_ERROR_GENERACIOINFORME
```

Label d'error: no es pot generar l'informe perquè no hi ha alarmes

---

## **TXT\_ERROR\_NOBLUETOOTH**

```
public static final java.lang.String TXT_ERROR_NOBLUETOOTH
```

Label d'error: no hi ha bluetooth

---

## **TXT\_ERROR\_NOSERVEIPPSTOP**

```
public static final java.lang.String TXT_ERROR_NOSERVEIPPSTOP
```

Label d'error: no s'ha trobat servei PPStop

---

## **TXT\_ERROR\_TRANSFERENCIAHISTORIAL**

```
public static final java.lang.String TXT_ERROR_TRANSFERENCIAHISTORIAL
```

Label d'error: error en la transferència d'historial

---

## **TXT\_ERROR\_NOPACIENT**

```
public static final java.lang.String TXT_ERROR_NOPACIENT
```

Label d'error: no hi ha cap pacient seleccionat

---

## **TXT\_ALERTA\_OPERACIOPOTTRIGAR**

```
public static final java.lang.String TXT_ALERTA_OPERACIOPOTTRIGAR
```

Label d'alerta: l'operació pot trigar uns segons

---

## **TXT\_ALERTA\_HISTORIALREBUT**

```
public static final java.lang.String TXT_ALERTA_HISTORIALREBUT
```

Label d'alerta: historial rebut

---

# Package Dades

Conté les classes per a gestionar i emmagatzemar totes les dades de l'aplicació

## Dades Class Alarma

```
java.lang.Object
|
+--Dades.Alarma
```

```
public class Alarma
extends java.lang.Object
```

Classe representativa d'una alarma

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
1.0 (Juny de 2010)

### Field Summary

private	<a href="#">CodiHistorial</a> CodiHistorial de l'alarma
private	<a href="#">DataAlarma</a> Data de l'alarma

### Constructor Summary

public	<a href="#">Alarma</a> (java.lang.String CodiHistorial, java.util.Date DataAlarma) Crea una instància d'Alarma
--------	---

### Method Summary

java.lang.String	<a href="#">getCodiHistorial</a> () Obte el codi de l'historial de l'alarma
java.util.Date	<a href="#">getDataAlarma</a> () Obte la data de l'alarma
java.lang.String	<a href="#">toString</a> () Representació textual de la classe Alarma, per a funcions de debugging

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### CodiHistorial

```
private java.lang.String CodiHistorial
```

CodiHistorial de l'alarma

---

## DataAlarma

```
private java.util.Date DataAlarma
```

Data de l'alarma

## Constructors

### Alarma

```
public Alarma(java.lang.String CodiHistorial,  
              java.util.Date DataAlarma)
```

Crea una instància d'Alarma

**Parameters:**

Timestamp - Timestamp de l'alarma

## Methods

### getCodiHistorial

```
public java.lang.String getCodiHistorial()
```

Obte el codi de l'historial de l'alarma

**Returns:**

Codi de l'historial de l'alarma

---

### getDataAlarma

```
public java.util.Date getDataAlarma()
```

Obte la data de l'alarma

**Returns:**

Data de l'alarma

---

### toString

```
public java.lang.String toString()
```

Representació textual de la classe Alarma, per a funcions de debugging

**Returns:**

Representacio textual de la classe



## Dades Class DB

```
java.lang.Object
|
+--Dades.DB
```

```
public class DB
extends java.lang.Object
```

Classe que permet interactuar amb la base de dades JavaDB(derby)

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
1.0 (Juny de 2010)

### Field Summary

private	<a href="#">C</a> Objecte de connexio a la base de dades
---------	---

### Constructor Summary

public	<a href="#">DB()</a> Crea una instància de DB; crea la base de dades i les taules si no existeixen
--------	---

### Method Summary

void	<a href="#">Connectar()</a> Estableix una connexió amb la base de dades
void	<a href="#">CrearTaules()</a> Crea les taules a la base de dades si aquestes no existeixen
void	<a href="#">DeleteAlarmes()</a> (java.lang.String condicio) Elimina totes les alarmes que compleixen una condició
void	<a href="#">DeletePacient()</a> (java.lang.String condicio) Elimina un pacient de la taula Pacients (i les files referenciades de la taula Alarmes)
void	<a href="#">Desconnectar()</a> Tanca la connexio amb la base de dades
void	<a href="#">InsertAlarma()</a> ( <a href="#">Alarma</a> Alarma) Insereix una alarma a la taula Alarmes
void	<a href="#">InsertPacient()</a> ( <a href="#">Pacient</a> Pacient) Insereix un pacient a la taula Pacients
java.util.List	<a href="#">SelectAlarmes()</a> (java.lang.String condicio) Fa una selecció d'alarmes de la taula Alarmes segons una condicio
java.util.List	<a href="#">SelectAlarmesPerMes()</a> (java.lang.String condicio) Genera el recompte d'alarmes mensual segons una condicio

int	<a href="#"><u>SelectIdioma()</u></a> Selecciona l'idioma de l'aplicació
java.lang.Integer	<a href="#"><u>SelectNumAlarmes()</u></a> (java.lang.String condicio) Genera el recompte d'alarmes donada una condicio
java.util.List	<a href="#"><u>SelectPacients()</u></a> (java.lang.String condicio) Fa una selecció de pacients de la taula Pacients segons una condicio
void	<a href="#"><u>UpdateIdioma()</u></a> (int idioma) Actualitza l'idioma de l'aplicació
void	<a href="#"><u>UpdatePacient()</u></a> ( <a href="#"><u>Pacient</u></a> Pacient, java.lang.String condicio) Actualitza un pacient de la taula Pacients

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### C

```
private java.sql.Connection C
```

Objecte de connexio a la base de dades

## Constructors

### DB

```
public DB()
```

Crea una instància de DB; crea la base de dades i les taules si no existeixen

## Methods

### Connectar

```
public void Connectar()
```

Estableix una connexió amb la base de dades

### Desconnectar

```
public void Desconnectar()
```

Tanca la connexio amb la base de dades

### CrearTaules

```
private void CrearTaules()  
throws java.sql.SQLException
```

Crea les taules a la base de dades si aquestes no existeixen

---

## SelectIdioma

```
public int SelectIdioma()
```

Selecciona l'idioma de l'aplicació

**Returns:**

Idioma de l'aplicacio

---

## UpdateIdioma

```
public void UpdateIdioma(int idioma)
```

Actualitza l'idioma de l'aplicació

**Parameters:**

idioma - Nou idioma de l'aplicació

---

## SelectPacients

```
public java.util.List SelectPacients(java.lang.String condicio)
```

Fa una selecció de pacients de la taula Pacients segons una condicio

**Parameters:**

condicio - Condicio de la selecció

**Returns:**

Llista de pacients seleccionats

---

## InsertPacient

```
public void InsertPacient(Pacient Pacient)
```

Insereix un pacient a la taula Pacients

**Parameters:**

Pacient - Pacient a inserir

---

## UpdatePacient

```
public void UpdatePacient(Pacient Pacient,  
java.lang.String condicio)
```

Actualitza un pacient de la taula Pacients

**Parameters:**

Pacient - Pacient a actualitzar

condicio - Condició de l'actualització

---

## DeletePacient

```
public void DeletePacient(java.lang.String condicio)
```

Elimina un pacient de la taula Pacients (i les files referenciades de la taula Alarmes)

**Parameters:**

Pacient - Pacient a eliminar

---

---

## SelectAlarmes

```
public java.util.List SelectAlarmes(java.lang.String condicio)
```

Fa una selecció d'alarmes de la taula Alarmes segons una condicio

**Parameters:**

condicio - Condicio de la selecció

**Returns:**

Llista d'alarmes seleccionades

---

## InsertAlarma

```
public void InsertAlarma(Alarma Alarma)
```

Insereix una alarma a la taula Alarmes

**Parameters:**

Alarma - Alarma a inserir

---

## DeleteAlarmest

```
public void DeleteAlarmest(java.lang.String condicio)
```

Elimina totes les alarmes que compleixen una condició

**Parameters:**

condicio - Condicio d'eliminació

---

## SelectAlarmesPerMes

```
public java.util.List SelectAlarmesPerMes(java.lang.String condicio)
```

Genera el recompte d'alarmes mensual segons una condicio

**Parameters:**

condicio - Condicio de la selecció

**Returns:**

Parella Mes - Num d'alarmes

---

## SelectNumAlarmes

```
public java.lang.Integer SelectNumAlarmes(java.lang.String condicio)
```

Genera el recompte d'alarmes donada una condicio

**Parameters:**

condicio - Condicio de la selecció

**Returns:**

Nombre d'alarmes

---

# Dades

## Class Historial

```
java.lang.Object
|
+--Dades.Historial
```

```
public class Historial
extends java.lang.Object
```

Classe representativa de l'Historial d'Alarmes

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juny de 2010)

### Field Summary

private	<a href="#"><u>TZid</u></a> Identificador de la TimeZone
private	<a href="#"><u>v</u></a> Vector de timestamps

### Constructor Summary

public	<a href="#"><u>Historial()</u></a> Crea una instància de Historial
--------	---

### Method Summary

void	<a href="#"><u>delHistorial()</u></a> Reseteja l'Historial d'Alarmes
void	<a href="#"><u>fromByteArray</u></a> (byte[] dades) Des-Serialitza l'objecte
java.util.Vector	<a href="#"><u>getHistorial()</u></a> Obté el vector amb l'Historial d'Alarmes
java.lang.String	<a href="#"><u>getTZid()</u></a> Obtenir l'identificador de la TimeZone
void	<a href="#"><u>setHistorial</u></a> (java.util.Vector valor) Guarda el vector amb l'Historial d'Alarmes
byte[]	<a href="#"><u>toByteArray</u></a> () Serialitza l'objecte

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### V

```
private java.util.Vector v
```

Vector de timestamps

---

### TZid

```
private java.lang.String tZid
```

Identificador de la TimeZone

---

## Constructors

### Historial

```
public Historial()
```

Crea una instància de Historial

---

## Methods

### getHistorial

```
public java.util.Vector getHistorial()
```

Obté el vector amb l'Historial d'Alarmes

**Returns:**

Vector de timestamps

---

### setHistorial

```
public void setHistorial(java.util.Vector valor)
```

Guarda el vector amb l'Historial d'Alarmes

**Parameters:**

valor - Vector de timestamps

---

### delHistorial

```
public void delHistorial()
```

Reseteja l'Historial d'Alarmes

---

### getTZid

```
public java.lang.String getTZid()
```

Obtenir l'identificador de la TimeZone

---

(continued from last page)

## toByteArray

```
public byte[] toByteArray()  
    throws java.io.IOException
```

Serialitza l'objecte

### Returns:

Stream de bytes (Objecte serialitzat)

---

## fromByteArray

```
public void fromByteArray(byte[] dades)  
    throws java.io.IOException
```

Des-Serialitza l'objecte

### Parameters:

dades - Stream de bytes (Objecte des-serialitzat)

## Dades Class Model

```
java.lang.Object
  |
  +- java.util.Observable
        |
        +- Dades.Model
```

```
public class Model
extends java.util.Observable
```

Classe que conté totes les dades de l'aplicació. Actua com a Model dins del patró MVC (Model-View-Controller)

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juny de 2010)

### Field Summary

private	<a href="#">DB</a> Objecte Gestor de la BD
private	<a href="#">Idioma</a> Idioma de la interfície de l'aplicació
private	<a href="#">PacientActual</a> Pacient que s'està analitzant
private	<a href="#">PacientsActuals</a> Llista de pacients donats d'alta

### Fields inherited from class java.util.Observable

changed, obs

### Constructor Summary

public	<a href="#">Model</a> () Crea una instància de Model
--------	---

### Method Summary

void	<a href="#">delPacient</a> ( <a href="#">Pacient</a> Pacient) Elimina un pacient (i les seves alarmes) existent en l'aplicació
java.util.List	<a href="#">getAlarmes</a> ( <a href="#">Pacient</a> Pacient, java.lang.String condicio) Obte les alarmes d'un pacient en concret, donada una condició
java.util.List	<a href="#">getDadesGBarresMensual</a> ( <a href="#">Pacient</a> Pacient)
java.util.List	<a href="#">getDadesGBarresPerHores</a> ( <a href="#">Pacient</a> Pacient)
java.util.List	<a href="#">getDadesGBarresUltimMes</a> ( <a href="#">Pacient</a> Pacient)



java.util.List	<a href="#"><u>getDadesGFormatgeDiurnNocturn</u></a> ( <a href="#"><u>Pacient</u></a> Pacient)
int	<a href="#"><u>getIdioma</u></a> () Obté l'idioma de la interfície de l'aplicació
<a href="#"><u>Pacient</u></a>	<a href="#"><u>getPacient</u></a> (java.lang.String CodiHistorial) Obte el pacient amb un cert codi d'historial
<a href="#"><u>Pacient</u></a>	<a href="#"><u>getPacientActual</u></a> () Obté el pacient que s'està analitzant en aquest moment
java.util.List	<a href="#"><u>getPacients</u></a> () Obte la llista de tots els pacients
void	<a href="#"><u>setAlarma</u></a> ( <a href="#"><u>Alarma</u></a> Alarma) Afegeix una nova alarma a l'aplicació
int	<a href="#"><u>setHistorial</u></a> ( <a href="#"><u>Pacient</u></a> P, <a href="#"><u>Historial</u></a> H) Insereix al sistema les alarmes contingudes en un objecte Historial
void	<a href="#"><u>setIdioma</u></a> (int Idioma) Estableix l'idioma de la interfície de l'aplicació
void	<a href="#"><u>setPacient</u></a> ( <a href="#"><u>Pacient</u></a> Pacient) Estableix un nou pacient a l'aplicació
void	<a href="#"><u>setPacientActual</u></a> ( <a href="#"><u>Pacient</u></a> PacientActual) Estableix el pacient que s'està analitzant en aquest moment
void	<a href="#"><u>updPacient</u></a> ( <a href="#"><u>Pacient</u></a> Pacient, java.lang.String CodiHistorial) Actualitza un pacient existent en l'aplicació

**Methods inherited from class** java.util.Observable

addObserver, clearChanged, countObservers, deleteObserver, deleteObservers, hasChanged, notifyObservers, notifyObservers, setChanged

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### DB

```
private Dades.DB DB
```

Objecte Gestor de la BD

### Idioma

```
private int Idioma
```

Idioma de la interfície de l'aplicació

## PacientsActuals

```
private java.util.List PacientsActuals
```

Llista de pacients donats d'alta

---

## PacientActual

```
private Dades.Pacient PacientActual
```

Pacient que s'està analitzant

## Constructors

### Model

```
public Model()
```

Crea una instància de Model

## Methods

### getIdioma

```
public int getIdioma()
```

Obté l'idioma de la interfície de l'aplicació

**Returns:**

Identificador de l'idioma

---

### setIdioma

```
public void setIdioma(int Idioma)
```

Estableix l'idioma de la interfície de l'aplicació

**Parameters:**

valor - Identificador de l'idioma

---

### getPacients

```
public java.util.List getPacients()
```

Obte la llista de tots els pacients

**Returns:**

Llista de pacients

---

### getPacient

```
public Pacient getPacient(java.lang.String CodiHistorial)
```

Obte el pacient amb un cert codi d'historial

**Parameters:**

CodiHistorial - Codi d'historial del pacient

---

(continued from last page)

**Returns:**

Pacient seleccionat

---

**setPacient**

```
public void setPacient(Pacient Pacient)
```

Estableix un nou pacient a l'aplicació

**Parameters:**

Pacient - Pacient a inserir

---

**updPacient**

```
public void updPacient(Pacient Pacient,  
    java.lang.String CodiHistorial)
```

Actualitza un pacient existent en l'aplicació

**Parameters:**

Pacient - Pacient a actualitzar

CodiHistorial - Codi d'historial del pacient a modificar

---

**delPacient**

```
public void delPacient(Pacient Pacient)
```

Elimina un pacient (i les seves alarmes) existent en l'aplicació

**Parameters:**

Pacient - Pacient a eliminar

---

**getAlarmes**

```
public java.util.List getAlarmes(Pacient Pacient,  
    java.lang.String condició)
```

Obte les alarmes d'un pacient en concret, donada una condició

**Parameters:**

Pacient - Pacient en qüestió

condicio - Condició de les alarmes a obtenir

---

**setAlarma**

```
public void setAlarma(Alarma Alarma)
```

Afegeix una nova alarma a l'aplicació

**Parameters:**

Alarma - Alarma a afegir

---

**getDadesGBarresMensual**

```
public java.util.List getDadesGBarresMensual(Pacient Pacient)
```

---

## getDadesGBarresUltimMes

```
public java.util.List getDadesGBarresUltimMes(Pacient Pacient)
```

---

## getDadesGBarresPerHores

```
public java.util.List getDadesGBarresPerHores(Pacient Pacient)
```

---

## getDadesGFormatgeDiurnNocturn

```
public java.util.List getDadesGFormatgeDiurnNocturn(Pacient Pacient)
```

---

## setPacientActual

```
public void setPacientActual(Pacient PacientActual)
```

Estableix el pacient que s'està analitzant en aquest moment

**Parameters:**

PacientActual - Pacient que s'està analitzant

---

## getPacientActual

```
public Pacient getPacientActual()
```

Obté el pacient que s'està analitzant en aquest moment

**Returns:**

Pacient que s'està analitzant

---

## setHistorial

```
public int setHistorial(Pacient P,  
                        Historial H)
```

Insereix al sistema les alarmes contingudes en un objecte Historial

**Parameters:**

Historial - Historial d'alarmes

**Returns:**

Nombre d'alarmes rebudes en l'historial

## Dades

### Class Pacient

```
java.lang.Object
  |
  +--Dades.Pacient
```

```
public class Pacient
  extends java.lang.Object
```

Classe representativa d'un pacient

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
1.0 (Juny de 2010)

### Field Summary

private	<a href="#"><u>CodiHistorial</u></a> Codi de l'historial PPStop del pacient
private	<a href="#"><u>Cognoms</u></a> Cognoms del pacient
private	<a href="#"><u>DarreraTransferencia</u></a> Data de la darrera transferència d'Historial
private	<a href="#"><u>DataAlta</u></a> Data d'alta del pacient
private	<a href="#"><u>DataNaixement</u></a> Data de naixement del pacient
private	<a href="#"><u>Nom</u></a> Nom del pacient
private	<a href="#"><u>Sexe</u></a> Sexe del pacient

### Constructor Summary

public	<a href="#"><u>Pacient</u></a> (java.lang.String CodiHistorial, java.lang.String Nom, java.lang.String Cognoms, java.util.Date DataNaixement, int Sexe, java.util.Date DataAlta, java.util.Date DarreraTransferencia) Crea una instància de Pacient
--------	--

### Method Summary

void	<a href="#"><u>ActualitzaDades</u></a> (java.lang.String CodiHistorial, java.lang.String Nom, java.lang.String Cognoms, java.util.Date DataNaixement, int Sexe) Estableix les dades del pacient
java.lang.String	<a href="#"><u>getCodiHistorial</u></a> () Obté el codi de l'historial PPStop del pacient

java.lang.String	<a href="#"><u>getCognoms()</u></a> Obté els cognoms del pacient
java.util.Date	<a href="#"><u>getDarreraTransferencia()</u></a> Obté la data de la darrera transferència d'historial
java.util.Date	<a href="#"><u>getDataAlta()</u></a> Obté la data d'alta del pacient
java.util.Date	<a href="#"><u>getDataNaixement()</u></a> Obté la data de naixement del pacient
java.lang.String	<a href="#"><u>getNom()</u></a> Obté el nom del pacient
int	<a href="#"><u>getSexe()</u></a> Obté el sexe del pacient
void	<a href="#"><u>setCodiHistorial()</u></a> (java.lang.String CodiHistorial) Estableix el codi de l'historial PPStop del pacient
void	<a href="#"><u>setCognoms()</u></a> (java.lang.String Cognoms) Estableix els cognoms del pacient
void	<a href="#"><u>setDarreraTransferencia()</u></a> (java.util.Date DarreraTransferencia) Estableix la data de la darrera transferència d'historial
void	<a href="#"><u>setDataAlta()</u></a> (java.util.Date DataAlta) Estableix la data d'alta del pacient
void	<a href="#"><u>setDataNaixement()</u></a> (java.util.Date DataNaixement) Estableix la data de naixement del pacient
void	<a href="#"><u>setNom()</u></a> (java.lang.String Nom) Estableix el nom del pacient
void	<a href="#"><u>setSexe()</u></a> (int Sexe) Estableix el sexe del pacient
java.lang.String	<a href="#"><u>toString()</u></a> Representació textual de la classe Pacient, per a funcions de debugging

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### CodiHistorial

```
private java.lang.String CodiHistorial
```

Codi de l'historial PPStop del pacient

(continued from last page)

## Nom

```
private java.lang.String Nom
```

Nom del pacient

## Cognoms

```
private java.lang.String Cognoms
```

Cognoms del pacient

## DataNaixement

```
private java.util.Date DataNaixement
```

Data de naixement del pacient

## Sexe

```
private int Sexe
```

Sexe del pacient

## DataAlta

```
private java.util.Date DataAlta
```

Data d'alta del pacient

## DarreraTransferencia

```
private java.util.Date DarreraTransferencia
```

Data de la darrera transferència d'Historial

## Constructors

### Pacient

```
public Pacient(java.lang.String CodiHistorial,  
               java.lang.String Nom,  
               java.lang.String Cognoms,  
               java.util.Date DataNaixement,  
               int Sexe,  
               java.util.Date DataAlta,  
               java.util.Date DarreraTransferencia)
```

Crea una instància de Pacient

#### Parameters:

CodiHistorial - Codi de l'historial PPStop del pacient

Nom - Nom del pacient

Cognoms - Cognoms del pacient

Edat - Edat del pacient

Sexe - Sexe del pacient

DataAlta - Data d'alta del pacient

DarreraTransferencia - Data de la darrera transferència d'Historial

(continued from last page)

## Methods

### ActualitzaDades

```
public void ActualitzaDades(java.lang.String CodiHistorial,  
    java.lang.String Nom,  
    java.lang.String Cognoms,  
    java.util.Date DataNaixement,  
    int Sexe)
```

Estableix les dades del pacient

**Parameters:**

CodiHistorial - Codi de l'historial PPStop del pacient

Nom - Nom del pacient

Cognoms - Cognoms del pacient

Edat - Edat del pacient

Sexe - Sexe del pacient

---

### setCodiHistorial

```
public void setCodiHistorial(java.lang.String CodiHistorial)
```

Estableix el codi de l'historial PPStop del pacient

**Parameters:**

CodiHistorial - Codi de l'historial PPStop del pacient

---

### getCodiHistorial

```
public java.lang.String getCodiHistorial()
```

Obté el codi de l'historial PPStop del pacient

**Returns:**

Codi de l'historial PPStop del pacient

---

### setNom

```
public void setNom(java.lang.String Nom)
```

Estableix el nom del pacient

**Parameters:**

Nom - Nom del pacient

---

### getNom

```
public java.lang.String getNom()
```

Obté el nom del pacient

**Returns:**

Nom del pacient

---

### setCognoms

```
public void setCognoms(java.lang.String Cognoms)
```



(continued from last page)

Estableix els cognoms del pacient

**Parameters:**

Cognoms - Cognoms del pacient

---

## getCognoms

```
public java.lang.String getCognoms()
```

Obté els cognoms del pacient

**Returns:**

Cognoms del pacient

---

## setDataNaixement

```
public void setDataNaixement(java.util.Date DataNaixement)
```

Estableix la data de naixement del pacient

**Parameters:**

DataNaixement - Data de naixement del pacient

---

## getDataNaixement

```
public java.util.Date getDataNaixement()
```

Obté la data de naixement del pacient

**Returns:**

Data de naixement del pacient

---

## setSexe

```
public void setSexe(int Sexe)
```

Estableix el sexe del pacient

**Parameters:**

Sexe - Sexe del pacient

---

## getSexe

```
public int getSexe()
```

Obté el sexe del pacient

**Returns:**

Sexe del pacient

---

## setDataAlta

```
public void setDataAlta(java.util.Date DataAlta)
```

Estableix la data d'alta del pacient

**Parameters:**

DataAlta - Data d'alta del pacient

---

---

## getDataAlta

```
public java.util.Date getDataAlta()
```

Obté la data d'alta del pacient

**Returns:**

Data d'alta del pacient

---

## setDarreraTransferencia

```
public void setDarreraTransferencia(java.util.Date DarreraTransferencia)
```

Estableix la data de la darrera transferència d'història

**Parameters:**

DarreraTransferencia - Data de la darrera transferència d'Història

---

## getDarreraTransferencia

```
public java.util.Date getDarreraTransferencia()
```

Obté la data de la darrera transferència d'història

**Returns:**

Data de la darrera transferència d'història

---

## toString

```
public java.lang.String toString()
```

Representació textual de la classe Pacient, per a funcions de debugging

**Returns:**

Representació textual de la classe

---

---

## Package Main

Conté la classe principal de l'aplicació

# Main Class App

```
java.lang.Object
|
+-Main.App
```

```
public class App
extends java.lang.Object
```

Classe principal de l'aplicació de seguiment PPStop

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
1.0 (Juny de 2010)

## Field Summary

private static	<a href="#">BT</a> Objecte d'obtenció d'historial a través de Bluetooth
private static	<a href="#">I</a> Objecte de generacio d'informe PDF
private static	<a href="#">M</a> Model de dades
private static	<a href="#">V</a> Graphic User Interface (GUI) de l'aplicació

## Constructor Summary

public	<a href="#">App()</a>
--------	-----------------------

## Method Summary

void	<a href="#">generarInforme</a> ( <a href="#">Pacient</a> P, java.lang.String path) Genera l'informe PDF del pacient especificat
static void	<a href="#">main</a> (java.lang.String[] args) Codi principal de l'aplicació
void	<a href="#">obtenirHistorial</a> ( <a href="#">Pacient</a> P) Obté l'historial d'alarmes del pacient especificat

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

(continued from last page)

## M

```
private static Dades.Model M
```

Model de dades

## V

```
private static Vista.GUI V
```

Graphic User Interface (GUI) de l'aplicació

## BT

```
private static Bluetooth.Bluetooth BT
```

Objecte d'obtenció d'historial a través de Bluetooth

## I

```
private static Utils.Informe I
```

Objecte de generació d'informe PDF

## Constructors

### App

```
public App()
```

## Methods

### obtenirHistorial

```
public void obtenirHistorial(Pacient P)
```

Obté l'historial d'alarmes del pacient especificat

#### Parameters:

P - Pacient del que es vol obtenir l'historial

### generarInforme

```
public void generarInforme(Pacient P,  
                           java.lang.String path)  
throws java.lang.Exception
```

Genera l'informe PDF del pacient especificat

#### Parameters:

P - Pacient del que es vol obtenir l'historial

path - Path on generar l'informe PDF

(continued from last page)

## **main**

```
public static void main(java.lang.String[] args)
```

Codi principal de l'aplicació

### **Parameters:**

args - Paràmetres de l'aplicació

---

# Package Utils

Conté les classes amb funcions auxiliars de certa utilitat

## Utils

### Class Grafics

```
java.lang.Object
  |
  +--Utils.Grafics
```

```
public class Grafics
extends java.lang.Object
```

Classe per a generar les gràfiques estadístiques dels pacients

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juny de 2010)

### Constructor Summary

public	<a href="#"><u>Grafics()</u></a> Crea una instància de Grafics
--------	---

### Method Summary

java.awt.image.Buffer edImage	<a href="#"><u>getGBarresDecimals</u></a> (java.util.List LR) Genera una grafica de barres amb les dades proporcionades (decimals)
java.awt.image.Buffer edImage	<a href="#"><u>getGBarresEnters</u></a> (java.util.List LR) Genera una grafica de barres amb les dades proporcionades (enters)
java.awt.image.Buffer edImage	<a href="#"><u>getGFormatge</u></a> (java.util.List LR) Genera una grafica circular amb les dades proporcionades

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Constructors

### Grafics

```
public Grafics()
```

Crea una instància de Grafics

## Methods

### getGBarresEnters

```
public java.awt.image.BufferedImage getGBarresEnters(java.util.List LR)
```

Genera una grafica de barres amb les dades proporcionades (enters)



(continued from last page)

**Parameters:**

LR - Llista de parells a representar

**Returns:**Imatge de la gràfica

---

**getGBarresDecimals**

```
public java.awt.image.BufferedImage getGBarresDecimals(java.util.List LR)
```

Genera una grafica de barres amb les dades proporcionades (decimals)

**Parameters:**

LR - Llista de parells a representar

**Returns:**Imatge de la gràfica

---

**getGFormatge**

```
public java.awt.image.BufferedImage getGFormatge(java.util.List LR)
```

Genera una grafica circular amb les dades proporcionades

**Parameters:**

LR - Llista de parells a representar

**Returns:**

Imatge de la gràfica

## Utils

### Class Informe

```
java.lang.Object
|
|--com.itextpdf.text.pdf.PdfPageEventHelper
|   |
|   |--Utils.Informe
```

#### All Implemented Interfaces:

com.itextpdf.text.pdf.PdfPageEvent

```
public class Informe
extends com.itextpdf.text.pdf.PdfPageEventHelper
```

Classe per a generar l'informe de seguiment PPStop en PDF

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

1.0 (Juny de 2010)

### Field Summary

private	<a href="#">FontNumPag</a> Font base per al numero de pàgina
private	<a href="#">M</a> Referència al Model
private	<a href="#">NumPag</a> Template del PDF per a inserir el número de pagina
private	<a href="#">Pacient</a> Pacient sobre el que es genera l'informe
private	<a href="#">PathGuardar</a> Path on guardar l'informe PDF generat

### Constructor Summary

public	<a href="#">Informe</a> ( <a href="#">Model</a> M, <a href="#">Pacient</a> Pacient, java.lang.String PathGuardar) Crea una instància d'Informe i genera l'informe PDF
--------	--

### Method Summary

void	<a href="#">crearHeader</a> (com.itextpdf.text.Document D) Escriu la capçalera del document
void	<a href="#">crearMetadata</a> (com.itextpdf.text.Document D) Escriu els metadata del document
void	<a href="#">crearTaulaAlarmes</a> (com.itextpdf.text.Document D, com.itextpdf.text.pdf.PdfWriter W) Escriu la taula d'alarmes del pacient al document

void	<a href="#"><code>crearTaulaDadesPacient</code></a> (com.itextpdf.text.Document D) Escriu la taula de dades del pacient al document
void	<a href="#"><code>crearTaulaGrafics</code></a> (com.itextpdf.text.Document D) Escriu la taula de gràfics estadístics del pacient al document
void	<a href="#"><code>generarInforme</code></a> () Genera l'informe PDF
void	<a href="#"><code>onCloseDocument</code></a> (com.itextpdf.text.pdf.PdfWriter W, com.itextpdf.text.Document D) Mètode que s'executa en acabar d'escriure tot el document
void	<a href="#"><code>onEndPage</code></a> (com.itextpdf.text.pdf.PdfWriter W, com.itextpdf.text.Document D) Mètode que s'executa en acabar d'escriure una pàgina del document
void	<a href="#"><code>onOpenDocument</code></a> (com.itextpdf.text.pdf.PdfWriter W, com.itextpdf.text.Document D) Mètode que s'executa en començar a escriure el document

#### Methods inherited from class com.itextpdf.text.pdf.PdfPageEventHelper

onChapter, onChapterEnd, onCloseDocument, onEndPage, onGenericTag, onOpenDocument, onParagraph, onParagraphEnd, onSection, onSectionEnd, onStartPage

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface com.itextpdf.text.pdf.PdfPageEvent

onChapter, onChapterEnd, onCloseDocument, onEndPage, onGenericTag, onOpenDocument, onParagraph, onParagraphEnd, onSection, onSectionEnd, onStartPage

## Fields

### M

private Dades.Model **M**

Referència al Model

### Pacient

private Dades.Pacient **Pacient**

Pacient sobre el que es genera l'informe

### PathGuardar

private java.lang.String **PathGuardar**

Path on guardar l'informe PDF generat

(continued from last page)

## NumPag

```
private com.itextpdf.text.pdf.PdfTemplate NumPag
```

Template del PDF per a inserir el número de pagina

## FontNumPag

```
private com.itextpdf.text.pdf.BaseFont FontNumPag
```

Font base per al numero de pàgina

## Constructors

### Informe

```
public Informe(Model M,  
               Pacient Pacient,  
               java.lang.String PathGuardar)
```

Crea una instància d'Informe i genera l'informe PDF

## Methods

### generarInforme

```
public void generarInforme()  
    throws java.lang.Exception
```

Genera l'informe PDF

### crearMetadata

```
private void crearMetadata(com.itextpdf.text.Document D)
```

Escriu els metadata del document

#### Parameters:

Document - Document de PDF

### crearHeader

```
private void crearHeader(com.itextpdf.text.Document D)  
    throws java.lang.Exception
```

Escriu la capçalera del document

#### Parameters:

Document - Document de PDF

### crearTaulaDadesPacient

```
private void crearTaulaDadesPacient(com.itextpdf.text.Document D)  
    throws java.lang.Exception
```

Escriu la taula de dades del pacient al document

#### Parameters:

Document - Document de PDF

---

## crearTaulaGrafics

```
private void crearTaulaGrafics(com.itextpdf.text.Document D)  
    throws java.lang.Exception
```

Escriu la taula de gràfics estadístics del pacient al document

### Parameters:

Document - Document de PDF

---

## crearTaulaAlarmes

```
private void crearTaulaAlarmes(com.itextpdf.text.Document D,  
    com.itextpdf.text.pdf.PdfWriter W)  
    throws java.lang.Exception
```

Escriu la taula d'alarmes del pacient al document

### Parameters:

Document - Document de PDF

PdfWriter - Escriptor de PDF

---

## onOpenDocument

```
public void onOpenDocument(com.itextpdf.text.pdf.PdfWriter W,  
    com.itextpdf.text.Document D)
```

Mètode que s'executa en començar a escriure el document

### Parameters:

PdfWriter - Escriptor de PDF

Document - Document de PDF

---

## onEndPage

```
public void onEndPage(com.itextpdf.text.pdf.PdfWriter W,  
    com.itextpdf.text.Document D)
```

Mètode que s'executa en acabar d'escriure una pàgina del document

### Parameters:

PdfWriter - Escriptor de PDF

Document - Document de PDF

---

## onCloseDocument

```
public void onCloseDocument(com.itextpdf.text.pdf.PdfWriter W,  
    com.itextpdf.text.Document D)
```

Mètode que s'executa en acabar d'escriure tot el document

### Parameters:

PdfWriter - Escriptor de PDF

Document - Document de PDF

---

# Utils

## Class Utils

```
java.lang.Object
|
+--Utils.Utils
```

```
public class Utils
extends java.lang.Object
```

Classe amb funcions útils d'àmbit general

**Author:**  
Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**  
1.0 (Juny de 2010)

### Constructor Summary

public	<a href="#">Utils()</a>
--------	-------------------------

### Method Summary

static java.util.Vector	<a href="#">invertirVector</a> (java.util.Vector vec) Inverteix els elements d'un vector
static java.util.Date	<a href="#">obtenirData</a> (long timestamp, java.lang.String TZid) Obté un objecte Date a partir d'un timestamp i un identificador de TimeZone
static java.util.Date	<a href="#">obtenirDataActual</a> () Obté un objecte Date amb la data actual
static java.lang.String	<a href="#">obtenirDataTXT</a> (java.util.Date date, int tipus, int idioma) Obté la data en un format de text determinat a partir d'un Date
static java.lang.String	<a href="#">obtenirDiferenciaData</a> (java.util.Date dataActual, java.util.Date dataAntiga, int idioma) Obté la diferència entre dues dates en format TXT
static int	<a href="#">obtenirEdat</a> (java.util.Date dataNaixement) Obté l'edat donada la data de naixement
static long	<a href="#">obtenirTimestampActual</a> () Obté el timestamp actual
static java.sql.Timestamp	<a href="#">obtenirTimestampSQL</a> (java.util.Date data) Obté un objecte Date amb la data actual
static java.lang.String	<a href="#">substituirStr</a> (java.lang.String text, java.lang.String comodi, java.lang.String substitut) Reemplaça strings d'una cadena de caràcters

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Constructors

### Utils

```
public Utils()
```

## Methods

### obtenerTimestampActual

```
public static long obtenerTimestampActual()
```

Obte el timestamps actual

**Returns:**

Timestamp actual

---

### obtenerDataActual

```
public static java.util.Date obtenerDataActual()
```

Obté un objecte Date amb la data actual

**Returns:**

Data actual

---

### obtenerData

```
public static java.util.Date obtenerData(long timestamp,  
java.lang.String TZid)
```

Obté un objecte Date a partir d'un timestamp i un identificador de TimeZone

**Parameters:**

timestamp - Timestamp a convertir

TZid - Identificador de la TimeZone

**Returns:**

Objecte Date

---

### obtenerTimestampSQL

```
public static java.sql.Timestamp obtenerTimestampSQL(java.util.Date data)
```

Obté un objecte Date amb la data actual

**Returns:**

Data actual

---

(continued from last page)

## obtenirDataTXT

```
public static java.lang.String obtenirDataTXT(java.util.Date date,
        int tipus,
        int idioma)
```

Obté la data en un format de text determinat a partir d'un Date

### Parameters:

timestamp - Timestamp a convertir

### Returns:

Data formatada

---

## obtenirDiferenciaData

```
public static java.lang.String obtenirDiferenciaData(java.util.Date dataActual,
        java.util.Date dataAntiga,
        int idioma)
```

Obté la diferència entre dues dates en format TXT

### Parameters:

timestamp - Timestamp de la data passada

idioma - Idioma desitjat

### Returns:

Diferència en format TXT

---

## obtenirEdat

```
public static int obtenirEdat(java.util.Date dataNaixement)
```

Obté l'edat donada la data de naixement

### Parameters:

dataNaixement - Data de naixement

### Returns:

Edat (anys)

---

## invertirVector

```
public static java.util.Vector invertirVector(java.util.Vector vec)
```

Inverteix els elements d'un vector

### Parameters:

vec - Vector a invertir

### Returns:

Vector invertit

---

## substituirStr

```
public static java.lang.String substituirStr(java.lang.String text,
        java.lang.String comodi,
        java.lang.String substitut)
```

Reemplaça strings d'una cadena de caràcters

---



(continued from last page)

**Parameters:**

`text` - Text sobre el qual s'aplica la substitució

`comodi` - Cadena de text a substituir

`substitut` - Cadena de text substituïda

**Returns:**

Text amb la substitució aplicada

# Utils

## Class UtilsVista

```
java.lang.Object
  |
  +--Utils.UtilsVista
```

```
public class UtilsVista
extends java.lang.Object
```

Classe amb funcions útils per a la GUI

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juny de 2010)

## Constructor Summary

public	<a href="#">UtilsVista()</a>
--------	------------------------------

## Method Summary

static java.awt.image.Buffer edImage	<a href="#">EscalarImatge</a> (java.awt.image.BufferedImage BI, int width, int height) Retorna una imatge escalada a les dimensions proporcionades
--	---

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Constructors

### UtilsVista

```
public UtilsVista()
```

## Methods

### EscalarImatge

```
public static java.awt.image.BufferedImage EscalarImatge( java.awt.image.BufferedImage
BI,
    int width,
    int height)
```

Retorna una imatge escalada a les dimensions proporcionades

**Parameters:**

BI - Imatge a escalar  
width - Amplada desitxada (px.)

(continued from last page)

height - Alçada desitxada (px.)

**Returns:**

Imatge escalada

---

# Package Vista

Conté les classes per a generar i gestionar la GUI (Graphic User Interface) de l'aplicació

# Vista

## Class DialogPacient

```

java.lang.Object
├── java.awt.Component
│   ├── java.awt.Container
│   │   ├── java.awt.Window
│   │   │   ├── java.awt.Dialog
│   │   │   │   ├── javax.swing.JDialog
│   │   │   │   └── Vista.DialogPacient

```

### All Implemented Interfaces:

[Vista](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.RootPaneContainer, javax.accessibility.Accessible, javax.swing.WindowConstants

### public class **DialogPacient**

extends javax.swing.JDialog

implements javax.swing.WindowConstants, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler, javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, [Vista](#)

Classe que conté tota la GUI per a donar d'alta un nou pacient o modificar-ne un de ja existent

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

1.0 (Juny de 2010)

## Field Summary

private	<a href="#">accio</a> Tipus d'acció a realitzar: crear nou pacient o modificar pacient que s'està tractant
private	<a href="#">BotoAltaModificar</a>
private	<a href="#">CaixaCodiHistorial</a>
private	<a href="#">CaixaCognoms</a>
private	<a href="#">CaixaDataNaixementAny</a>
private	<a href="#">CaixaDataNaixementDia</a>
private	<a href="#">CaixaDataNaixementMes</a>
private	<a href="#">CaixaNom</a>

private	<a href="#">LabelCodiHistorial</a>
private	<a href="#">LabelCognoms</a>
private	<a href="#">LabelDataNaixement</a>
private	<a href="#">LabelFormatDataNaixement</a>
private	<a href="#">LabelNom</a>
private	<a href="#">LabelSexe</a>
private	<a href="#">LabelText</a>
private	<a href="#">M</a> Referència al model de dades
private	<a href="#">SelectorSexe</a>

**Fields inherited from class `javax.swing.JDialog`**

`accessibleContext`, `defaultCloseOperation`, `defaultLookAndFeelDecoratedKey`, `rootPane`, `rootPaneCheckingEnabled`, `transferHandler`

**Fields inherited from class `java.awt.Dialog`**

`base`, `blockedWindows`, `DEFAULT_MODAL_EXCLUSION_TYPE`, `DEFAULT_MODALITY_TYPE`, `isInDispose`, `isInHide`, `keepBlocking`, `modal`, `modalDialogs`, `modalFilter`, `modalityType`, `nameCounter`, `resizable`, `serialVersionUID`, `showAppContext`, `title`, `undecorated`

**Fields inherited from class `java.awt.Window`**

`allWindows`, `alwaysOnTop`, `anchor`, `base`, `beforeFirstShow`, `beforeFirstWindowShown`, `focusableWindowState`, `focusMgr`, `icons`, `inputContext`, `inputContextLock`, `isInShow`, `isTrayIconWindow`, `locationByPlatform`, `locationByPlatformProp`, `log`, `modalBlocker`, `modalExclusionType`, `nameCounter`, `opacity`, `opaque`, `OPENED`, `ownedWindowList`, `securityWarningAlignmentX`, `securityWarningAlignmentY`, `securityWarningHeight`, `securityWarningPointX`, `securityWarningPointY`, `securityWarningWidth`, `serialVersionUID`, `shape`, `showWithParent`, `state`, `syncLWRequests`, `systemSyncLWRequests`, `temporaryLostComponent`, `TRANSPARENT_BACKGROUND_COLOR`, `warningString`, `weakThis`, `windowFocusListener`, `windowListener`, `windowSerializedDataVersion`, `windowStateListener`

**Fields inherited from class `java.awt.Container`**

`component`, `containerListener`, `containerSerializedDataVersion`, `dbg`, `descendantsCount`, `dispatcher`, `EMPTY_ARRAY`, `focusCycleRoot`, `focusTraversalPolicy`, `focusTraversalPolicyProvider`, `INCLUDE_SELF`, `layoutMgr`, `listeningBoundsChildren`, `listeningChildren`, `mixingLog`, `modalAppContext`, `modalComp`, `numOfHWComponents`, `numOfLWComponents`, `preserveBackgroundColor`, `printing`, `printingThreads`, `SEARCH_HEAVYWEIGHTS`, `serialPersistentFields`, `serialVersionUID`

**Fields inherited from class `java.awt.Component`**

accessibleContext, actionListenerK, adjustmentListenerK, appContext, background, backgroundEraseDisabled, BOTTOM\_ALIGNMENT, boundsOp, bufferStrategy, CENTER\_ALIGNMENT, changeSupport, coalesceEventsParams, coalesceMap, coalescingEnabled, componentListener, componentListenerK, componentOrientation, componentSerializedDataVersion, compoundShape, containerListenerK, cursor, dbg, dropTarget, enabled, eventCache, eventMask, FOCUS\_TRAVERSABLE\_DEFAULT, FOCUS\_TRAVERSABLE\_SET, FOCUS\_TRAVERSABLE\_UNKNOWN, focusable, focusListener, focusListenerK, focusLog, focusTraversalKeyPropertyNames, focusTraversalKeys, focusTraversalKeysEnabled, font, foreground, graphicsConfig, height, hierarchyBoundsListener, hierarchyBoundsListenerK, hierarchyListener, hierarchyListenerK, ignoreRepaint, incRate, inputMethodListener, inputMethodListenerK, isAddNotifyComplete, isFocusTraversableOverridden, isInc, isOpaqueForMixing, isPacked, itemListenerK, keyListener, keyListenerK, LEFT\_ALIGNMENT, locale, LOCK, log, maxSize, maxSizeSet, minSize, minSizeSet, mixingLog, mouseListener, mouseListenerK, mouseMotionListener, mouseMotionListenerK, mouseWheelListener, mouseWheelListenerK, name, nameExplicitlySet, nativeInLightFixer, newEventsOnly, objectLock, opaquePropertyChangeListener, ownedWindowK, parent, peer, peerFont, popups, prefSize, prefSizeSet, privateKey, requestFocusController, RIGHT\_ALIGNMENT, serialVersionUID, textListenerK, TOP\_ALIGNMENT, valid, visible, width, windowClosingException, windowFocusListenerK, windowListenerK, windowStateListenerK, x, y

#### Fields inherited from interface `java.awt.image.ImageObserver`

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

#### Fields inherited from interface `javax.swing.WindowConstants`

DISPOSE\_ON\_CLOSE, DO\_NOTHING\_ON\_CLOSE, EXIT\_ON\_CLOSE, HIDE\_ON\_CLOSE

## Constructor Summary

public	<a href="#">DialogPacient</a> ( <code>javax.swing.JFrame</code> GUI, <a href="#">Model</a> M, int accio, <code>java.lang.String</code> titol) Crea una nova instància de DialogPacient
--------	---

## Method Summary

void	<a href="#">BotoAltaModificarActionPerformed</a> ( <code>java.awt.event.ActionEvent</code> evt) Conté les accions a dur a terme en pulsar el botó
boolean	<a href="#">comprovarCamps</a> () Comprova que els camps hagin estat omplerts correctament
void	<a href="#">initComponents</a> ()
void	<a href="#">initValors</a> () Inicialitza les variables del panell
void	<a href="#">mostrarError</a> ( <code>java.lang.String</code> textError) Mostra un missatge d'error per pantalla
void	<a href="#">update</a> ( <code>java.util.Observable</code> o, <code>java.lang.Object</code> arg) Actualitza la vista amb les noves dades del Model (patró MVC)

#### Methods inherited from class `javax.swing.JDialog`

```
addImpl, createRootPane, dialogInit, getAccessibleContext, getContentPane,
getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane,
getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated,
isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, repaint,
setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated,
setGlassPane, setJMenuBar, setLayeredPane, setLayout, setRootPane,
setRootPaneCheckingEnabled, setTransferHandler, update
```

#### Methods inherited from class `java.awt.Dialog`

```
addNotify, blockWindow, checkShouldBeBlocked, conditionalShow,
constructComponentName, doDispose, getAccessibleContext, getModalityType, getTitle,
hide, hideAndDisposeHandler, hideAndDisposePreHandler, initIDs, interruptBlocking,
isModal_NoClientCode, isModal, isResizable, isUndecorated, modalHide, modalityPopped,
modalityPushed, modalShow, paramString, readObject, setModal, setModalityType,
setResizable, setTitle, setUndecorated, setVisible, shouldBlock, show, toBack,
unblockWindow
```

#### Methods inherited from class `java.awt.Window`

```
addNotify, addOwnedWindow, addPropertyChangeListener, addPropertyChangeListener,
addToWindowList, addWindowFocusListener, addWindowListener, addWindowStateListener,
adjustDescendantsOnParent, adjustListeningChildrenOnParent, applyCompoundShape,
applyCurrentShape, applyOpacity, applyOpaque, applyResourceBundle,
applyResourceBundle, applyShape, calculateSecurityWarningPosition,
canContainFocusOwner, clearMostRecentFocusOwnerOnHide, connectOwnedWindow,
constructComponentName, createBufferStrategy, createBufferStrategy,
deliverMouseWheelToAncestor, deserializeResources, dispatchEventImpl,
dispatchMouseWheelToAncestor, dispose, disposeImpl, doDispose, doesClassImplement,
doesImplement, eventEnabled, getAccessibleContext, getAllUnblockedWindows,
getAllWindows, getBufferStrategy, getContainer, getDocumentRoot,
getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner,
getFocusTraversalKeys, getGraphicsConfiguration, getIconImages, getInputContext,
getListeners, getLocale, getLocationOnWindow, getModalBlocker, getModalExclusionType,
getMostRecentFocusOwner, getOwnedWindows_NoClientCode, getOwnedWindows,
getOwner_NoClientCode, getOwner, getOwnerlessWindows, getTemporaryLostComponent,
getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners,
getWindows, getWindows, getWindowStateListeners, hide, init, initDeserializedWindow,
initIDs, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isFocusableWindow,
isFocusCycleRoot, isFocused, isLocationByPlatform, isModalBlocked, isModalExcluded,
isRecursivelyVisible, isShowing, mixOnReshaping, ownedInit, pack, postEvent,
postProcessKeyEvent, postWindowEvent, preProcessKeyEvent, processEvent,
processWindowEvent, processWindowFocusEvent, processWindowStateEvent, readObject,
removeFromWindowList, removeFromWindowList, removeNotify, removeOwnedWindow,
removeWindowFocusListener, removeWindowListener, removeWindowStateListener, resetGC,
reshape, setAlwaysOnTop, setBounds, setBounds, setClientSize, setCursor,
setFocusableWindowState, setFocusCycleRoot, setIconImage, setIconImages,
setLayersOpaque, setLocationByPlatform, setLocationRelativeTo, setMinimumSize,
setModalBlocked, setModalExclusionType, setSize, setSize, setTemporaryLostComponent,
setVisible, setWarningString, show, toBack_NoClientCode, toBack,
toFront_NoClientCode, toFront, updateChildFocusableWindowState,
updateChildrenBlocking, updateWindow, writeObject
```

#### Methods inherited from class `java.awt.Container`



```

add, add, add, add, add, addContainerListener, addDelicately, addImpl, addNotify,
addPropertyChangeListener, addPropertyChangeListener, adjustDecendantsOnParent,
adjustDescendants, adjustListeningChildren, applyComponentOrientation,
areFocusTraversalKeysSet, canContainFocusOwner, checkAdding, checkAddToSelf, checkGD,
checkNotAWindow, clearCurrentFocusCycleRootOnHide, clearMostRecentFocusOwnerOnHide,
containsFocus, countComponents, countHierarchyMembers, createChildHierarchyEvents,
createHierarchyEvents, decreaseComponentCount, deliverEvent, dispatchEventImpl,
dispatchEventToSelf, doLayout, eventEnabled, findComponentAt, findComponentAt,
findComponentAt, findComponentAtImpl, findTraversalRoot, getAccessibleAt,
getAccessibleChild, getAccessibleChildrenCount, getAlignmentX, getAlignmentY,
getBottommostComponentIndex, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents_NoClientCode, getComponents, getComponentZOrder,
getContainerListeners, getDropTargetEventTarget, getFocusTraversalKeys,
getFocusTraversalPolicy, getHeavyweightContainer, getInsets, getLayout, getListeners,
getListenersCount, getMaximumSize, getMinimumSize, getMouseEventTarget,
getMouseEventTarget, getMouseEventTargetImpl, getMousePosition, getOpaqueShape,
getPreferredSize, getTopmostComponentIndex, hasHeavyweightDescendants,
hasLightweightDescendants, increaseComponentCount, initializeFocusTraversalKeys,
initIDs, insets, invalidate, invalidateTree, isAncestorOf, isFocusCycleRoot,
isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet,
isParentOf, isRemoveNotifyNeeded, isSameOrAncestorOf, layout, lightweightPaint,
lightweightPrint, list, list, locate, minimumSize, mixOnShowing, mixOnValidating,
mixOnZOrderChanging, nextFocusHelper, numListening, paint, paintComponents,
paintHeavyweightComponents, paramString, postProcessKeyEvent, postsOldMouseEvents,
preferredSize, preProcessKeyEvent, print, printComponents,
printHeavyweightComponents, processContainerEvent, processEvent, proxyEnableEvents,
readObject, recursiveApplyCurrentShape, recursiveApplyCurrentShape,
recursiveApplyCurrentShape, recursiveSubtractAndApplyShape,
recursiveSubtractAndApplyShape, recursiveSubtractAndApplyShape, remove, remove,
removeAll, removeContainerListener, removeDelicately, removeNotify, reparentChild,
reparentTraverse, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout,
startLWModal, stopLWModal, transferFocusBackward, transferFocusDownCycle, update,
validate, validateTree, writeObject

```

**Methods inherited from class** java.awt.Component



```
removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener,
removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removeNotify, removePropertyChangeListener,
removePropertyChangeListener, repaint, repaint, repaint, repaint,
repaintParentIfNeeded, requestFocus, requestFocus, requestFocus, requestFocus,
requestFocusHelper, requestFocusHelper, requestFocusInWindow, requestFocusInWindow,
requestFocusInWindow, requestFocusInWindow, resetGC, reshape, reshapeNativePeer,
resize, resize, setBackground, setBounds, setBounds, setBoundsOp,
setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeys_NoIDCheck, setFocusTraversalKeys, setFocusTraversalKeysEnabled,
setFont, setForeground, setGCFromPeer, setIgnoreRepaint, setLocale, setLocation,
setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setRequestFocusController, setSize, setSize, setVisible, show, show, size,
subtractAndApplyShape, toString, transferFocus, transferFocusBackward,
transferFocusUpCycle, update, updateCursorImmediately, validate, writeObject
```

#### Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

#### Methods inherited from interface `java.awt.image.ImageObserver`

```
imageUpdate
```

#### Methods inherited from interface `java.awt.MenuContainer`

```
getFont, postEvent, remove
```

#### Methods inherited from interface `javax.accessibility.Accessible`

```
getAccessibleContext
```

#### Methods inherited from interface `javax.accessibility.Accessible`

```
getAccessibleContext
```

#### Methods inherited from interface `javax.swing.RootPaneContainer`

```
getContentPane, getGlassPane, getLayeredPane, getRootPane, setContentPane,
setGlassPane, setLayeredPane
```

#### Methods inherited from interface `javax.swing.TransferHandler.HasGetTransferHandler`

```
getTransferHandler
```

#### Methods inherited from interface `java.util.Observer`

```
update
```

## Fields

(continued from last page)

---

## M

```
private Dades.Model M
```

Referència al model de dades

---

## accio

```
private int accio
```

Tipus d'acció a realitzar: crear nou pacient o modificar pacient que s'està tractant

---

## BotoAltaModificar

```
private javax.swing.JButton BotoAltaModificar
```

---

## CaixaCodiHistorial

```
private javax.swing.JTextField CaixaCodiHistorial
```

---

## CaixaCognoms

```
private javax.swing.JTextField CaixaCognoms
```

---

## CaixaDataNaixementAny

```
private javax.swing.JTextField CaixaDataNaixementAny
```

---

## CaixaDataNaixementDia

```
private javax.swing.JTextField CaixaDataNaixementDia
```

---

## CaixaDataNaixementMes

```
private javax.swing.JTextField CaixaDataNaixementMes
```

---

## CaixaNom

```
private javax.swing.JTextField CaixaNom
```

---

## LabelCodiHistorial

```
private javax.swing.JLabel LabelCodiHistorial
```

---

(continued from last page)

---

## LabelCognoms

```
private javax.swing.JLabel LabelCognoms
```

---

## LabelDataNaixement

```
private javax.swing.JLabel LabelDataNaixement
```

---

## LabelFormatDataNaixement

```
private javax.swing.JLabel LabelFormatDataNaixement
```

---

## LabelNom

```
private javax.swing.JLabel LabelNom
```

---

## LabelSexe

```
private javax.swing.JLabel LabelSexe
```

---

## LabelText

```
private javax.swing.JLabel LabelText
```

---

## SelectorSexe

```
private java.awt.Choice SelectorSexe
```

---

## Constructors

### DialogPacient

```
public DialogPacient(javax.swing.JFrame GUI,  
                    Model M,  
                    int accio,  
                    java.lang.String titol)
```

Crea una nova instància de DialogPacient

#### Parameters:

GUI - Referència a la GUI principal

M - Referència al Model de dades

(continued from last page)

accio - Indica si es tracta d'una alta de pacient o una modificació de dades de pacient

titol - Títol de la finestra

## Methods

### comprovarCamps

```
private boolean comprovarCamps()
```

Comprova que els camps hagin estat omplerts correctament

### mostrarError

```
private void mostrarError(java.lang.String textError)
```

Mostra un missatge d'error per pantalla

**Parameters:**

textError - Text del missatge d'error

### initComponents

```
private void initComponents()
```

### BotoAltaModificarActionPerformed

```
private void BotoAltaModificarActionPerformed(java.awt.event.ActionEvent evt)
```

Conté les accions a dur a terme en polsar el botó

**Parameters:**

evt - Event generat sobre el botó

### initValors

```
public void initValors()
```

Inicialitza les variables del panell

### update

```
public void update(java.util.Observable o,  
java.lang.Object arg)
```

Actualitza la vista amb les noves dades del Model (patró MVC)

**Parameters:**

o - Observable que notifica el canvi en el Model

arg - Arguments

## Vista Class GUI

```

java.lang.Object
├-- java.awt.Component
│   ├── java.awt.Container
│   │   ├── java.awt.Window
│   │   │   ├── java.awt.Frame
│   │   │   │   ├── javax.swing.JFrame
│   │   │   │   └-- Vista.GUI

```

### All Implemented Interfaces:

[Vista](#), java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible, java.awt.MenuContainer, javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.RootPaneContainer, javax.accessibility.Accessible, javax.swing.WindowConstants

### public class GUI

extends javax.swing.JFrame

implements javax.swing.WindowConstants, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler, java.awt.MenuContainer, javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, [Vista](#)

Classe que conté tota la gestió de la GUI (Graphic User Interface). Actua com a Vista dins del patró MVC (Model-View-Controller)

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

1.0 (Juny de 2010)

Field Summary		
private	<a href="#">A</a>	Referència a l'App
private	<a href="#">BG</a>	Element gràfic de la GUI: agrupació de botons
private	<a href="#">DialogPacient</a>	Element gràfic de la GUI:
private	<a href="#">Grafic1</a>	
private	<a href="#">jPanell1</a>	
private	<a href="#">jScrollPane1</a>	
private	<a href="#">jSeparator1</a>	
private	<a href="#">LabelCodiHistorial</a>	

private	<a href="#"><u>LabelCodiHistorialAns</u></a>
private	<a href="#"><u>LabelDarreraTransferencia</u></a>
private	<a href="#"><u>LabelDarreraTransferenciaAns</u></a>
private	<a href="#"><u>LabelDataAlta</u></a>
private	<a href="#"><u>LabelDataAltaAns</u></a>
private	<a href="#"><u>LabelEdat</u></a>
private	<a href="#"><u>LabelEdatAns</u></a>
private	<a href="#"><u>LabelSeleccionarGrafic</u></a>
private	<a href="#"><u>LabelSeleccionarPacient</u></a>
private	<a href="#"><u>LabelSexe</u></a>
private	<a href="#"><u>LabelSexeAns</u></a>
private	<a href="#"><u>M</u></a> Referència al model de dades
private	<a href="#"><u>MB</u></a> Element gràfic de la GUI: barra de menú
private	<a href="#"><u>MenuAjuda</u></a> Element gràfic de la GUI: menu ajuda
private	<a href="#"><u>MenuArxiu</u></a> Element gràfic de la GUI: menu arxiu
private	<a href="#"><u>MenuIdioma</u></a> Element gràfic de la GUI: menu idioma
private	<a href="#"><u>MenuItemEliminarPacient</u></a> Element gràfic de la GUI: menu ítem eliminar pacient
private	<a href="#"><u>MenuItemGenerarInformePDF</u></a> Element gràfic de la GUI: menu ítem generar informe PDF
private	<a href="#"><u>MenuItemIdiomaAngles</u></a> Element gràfic de la GUI: menu ítem idioma anglès
private	<a href="#"><u>MenuItemIdiomaCastella</u></a> Element gràfic de la GUI: menu ítem idioma castellà
private	<a href="#"><u>MenuItemIdiomaCatala</u></a> Element gràfic de la GUI: menu ítem idioma català
private	<a href="#"><u>MenuItemModificarPacient</u></a> Element gràfic de la GUI: menu ítem modificar pacient



private	<a href="#">MenuItemMostrarAjuda</a> Element gràfic de la GUI: menu ítem mostrar ajuda
private	<a href="#">MenuItemNouPacient</a> Element gràfic de la GUI: menu ítem nou pacient
private	<a href="#">MenuItemObtenirHistorial</a> Element gràfic de la GUI: menu ítem obtenir historial
private	<a href="#">MenuItemSortir</a> Element gràfic de la GUI: menu ítem sortir
private	<a href="#">SelectorGrafic</a>
private	<a href="#">SelectorPacient</a>
private	<a href="#">TaulaAlarmes</a>

#### Fields inherited from class javax.swing.JFrame

accessibleContext, defaultCloseOperation, defaultLookAndFeelDecoratedKey, EXIT\_ON\_CLOSE, rootPane, rootPaneCheckingEnabled, transferHandler

#### Fields inherited from class java.awt.Frame

base, CROSSHAIR\_CURSOR, DEFAULT\_CURSOR, E\_RESIZE\_CURSOR, frameSerializedDataVersion, HAND\_CURSOR, ICONIFIED, MAXIMIZED\_BOTH, MAXIMIZED\_HORIZ, MAXIMIZED\_VERT, maximizedBounds, mbManagement, menuBar, MOVE\_CURSOR, N\_RESIZE\_CURSOR, nameCounter, NE\_RESIZE\_CURSOR, NORMAL, NW\_RESIZE\_CURSOR, ownedWindows, resizable, S\_RESIZE\_CURSOR, SE\_RESIZE\_CURSOR, serialVersionUID, state, SW\_RESIZE\_CURSOR, TEXT\_CURSOR, title, undecorated, W\_RESIZE\_CURSOR, WAIT\_CURSOR

#### Fields inherited from class java.awt.Window

allWindows, alwaysOnTop, anchor, base, beforeFirstShow, beforeFirstWindowShown, focusableWindowState, focusMgr, icons, inputContext, inputContextLock, isInShow, isTrayIconWindow, locationByPlatform, locationByPlatformProp, log, modalBlocker, modalExclusionType, nameCounter, opacity, opaque, OPENED, ownedWindowList, securityWarningAlignmentX, securityWarningAlignmentY, securityWarningHeight, securityWarningPointX, securityWarningPointY, securityWarningWidth, serialVersionUID, shape, showWithParent, state, syncLWRequests, systemSyncLWRequests, temporaryLostComponent, TRANSPARENT\_BACKGROUND\_COLOR, warningString, weakThis, windowFocusListener, windowListener, windowSerializedDataVersion, windowStateListener

#### Fields inherited from class java.awt.Container

component, containerListener, containerSerializedDataVersion, dbg, descendantsCount, dispatcher, EMPTY\_ARRAY, focusCycleRoot, focusTraversalPolicy, focusTraversalPolicyProvider, INCLUDE\_SELF, layoutMgr, listeningBoundsChildren, listeningChildren, mixingLog, modalAppContext, modalComp, numOfHWComponents, numOfLWComponents, preserveBackgroundColor, printing, printingThreads, SEARCH\_HEAVYWEIGHTS, serialPersistentFields, serialVersionUID

#### Fields inherited from class java.awt.Component

accessibleContext, actionListenerK, adjustmentListenerK, appContext, background, backgroundEraseDisabled, BOTTOM\_ALIGNMENT, boundsOp, bufferStrategy, CENTER\_ALIGNMENT, changeSupport, coalesceEventsParams, coalesceMap, coalescingEnabled, componentListener, componentListenerK, componentOrientation, componentSerializedDataVersion, compoundShape, containerListenerK, cursor, dbg, dropTarget, enabled, eventCache, eventMask, FOCUS\_TRAVERSABLE\_DEFAULT, FOCUS\_TRAVERSABLE\_SET, FOCUS\_TRAVERSABLE\_UNKNOWN, focusable, focusListener, focusListenerK, focusLog, focusTraversalKeyPropertyNames, focusTraversalKeys, focusTraversalKeysEnabled, font, foreground, graphicsConfig, height, hierarchyBoundsListener, hierarchyBoundsListenerK, hierarchyListener, hierarchyListenerK, ignoreRepaint, incRate, inputMethodListener, inputMethodListenerK, isAddNotifyComplete, isFocusTraversableOverridden, isInc, isOpaqueForMixing, isPacked, itemListenerK, keyListener, keyListenerK, LEFT\_ALIGNMENT, locale, LOCK, log, maxSize, maxSizeSet, minSize, minSizeSet, mixingLog, mouseListener, mouseListenerK, mouseMotionListener, mouseMotionListenerK, mouseWheelListener, mouseWheelListenerK, name, nameExplicitlySet, nativeInLightFixer, newEventsOnly, objectLock, opaquePropertyChangeListener, ownedWindowK, parent, peer, peerFont, popups, prefSize, prefSizeSet, privateKey, requestFocusController, RIGHT\_ALIGNMENT, serialVersionUID, textListenerK, TOP\_ALIGNMENT, valid, visible, width, windowClosingException, windowFocusListenerK, windowListenerK, windowStateListenerK, x, y

#### Fields inherited from interface `java.awt.image.ImageObserver`

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

#### Fields inherited from interface `javax.swing.WindowConstants`

DISPOSE\_ON\_CLOSE, DO\_NOTHING\_ON\_CLOSE, EXIT\_ON\_CLOSE, HIDE\_ON\_CLOSE

## Constructor Summary

public	<a href="#">GUI</a> ( <a href="#">App</a> A, <a href="#">Model</a> M) Crea una nova instància de GUI
--------	---

## Method Summary

void	<a href="#">actualitzaDadesPacient</a> () Actualitza les dades del pacient seleccionat a la GUI
void	<a href="#">actualitzaGrafic</a> () Actualitza el gràfic d'alarmes de la GUI
void	<a href="#">actualitzaParametres</a> () Actualitza els paràmetres de la GUI
void	<a href="#">actualitzaTaula</a> () Actualitza la taula d'alarmes de la GUI
void	<a href="#">crearMenu</a> () Crea el menú de l'aplicació
void	<a href="#">initComponents</a> () This method is called from within the constructor to initialize the form.
void	<a href="#">mostrarAlerta</a> (java.lang.String text, int tipus) Mostra un missatge d'alerta d'un tipus determinat (informació, error...)
int	<a href="#">mostrarConfirmacio</a> (java.lang.String textConfirmacio) Mostra un missatge de confirmació per pantalla

void	<a href="#"><code>mostrarDialogPacient</code></a> (int accio, java.lang.String titol) Mostra la finestra de creació/modificació d'un pacient
java.lang.String	<a href="#"><code>seleccionaPath</code></a> (java.lang.String nomfitxer) Mostra una finestra selectora de ruta i nom de fitxer
void	<a href="#"><code>SelectorGraficListener</code></a> (java.awt.event.ItemEvent evt)
void	<a href="#"><code>SelectorPacientListener</code></a> (java.awt.event.ItemEvent evt)
void	<a href="#"><code>update</code></a> (java.util.Observable o, java.lang.Object arg) Actualitza la vista amb les noves dades del Model (patró MVC)

#### Methods inherited from class javax.swing.JFrame

`addImpl`, `createRootPane`, `frameInit`, `getAccessibleContext`, `getContentPane`, `getDefaultCloseOperation`, `getGlassPane`, `getGraphics`, `getJMenuBar`, `getLayeredPane`, `getRootPane`, `getTransferHandler`, `isDefaultLookAndFeelDecorated`, `isRootPaneCheckingEnabled`,  `paramString`, `processWindowEvent`, `remove`, `repaint`, `setContentPane`, `setDefaultCloseOperation`, `setDefaultLookAndFeelDecorated`, `setGlassPane`, `setIconImage`, `setJMenuBar`, `setLayeredPane`, `setLayout`, `setRootPane`, `setRootPaneCheckingEnabled`, `setTransferHandler`, `update`

#### Methods inherited from class java.awt.Frame

`addNotify`, `constructComponentName`, `getAccessibleContext`, `getCursorType`, `getExtendedState`, `getFrames`, `getIconImage`, `getMaximizedBounds`, `getMenuBar`, `getState`, `getTitle`, `init`, `initIDs`, `isFrameStateSupported`, `isResizable`, `isUndecorated`,  `paramString`, `postProcessKeyEvent`, `readObject`, `remove`, `removeNotify`, `setCursor`, `setExtendedState`, `setIconImage`, `setMaximizedBounds`, `setMenuBar`, `setResizable`, `setState`, `setTitle`, `setUndecorated`, `writeObject`

#### Methods inherited from class java.awt.Window

```

addNotify, addOwnedWindow, addPropertyChangeListener, addPropertyChangeListener,
addToWindowList, addWindowFocusListener, addWindowListener, addWindowStateListener,
adjustDecendantsOnParent, adjustListeningChildrenOnParent, applyCompoundShape,
applyCurrentShape, applyOpacity, applyOpaque, applyResourceBundle,
applyResourceBundle, applyShape, calculateSecurityWarningPosition,
canContainFocusOwner, clearMostRecentFocusOwnerOnHide, connectOwnedWindow,
constructComponentName, createBufferStrategy, createBufferStrategy,
deliverMouseWheelToAncestor, deserializeResources, dispatchEventImpl,
dispatchMouseWheelToAncestor, dispose, disposeImpl, doDispose, doesClassImplement,
doesImplement, eventEnabled, getAccessibleContext, getAllUnblockedWindows,
getAllWindows, getBufferStrategy, getContainer, getDocumentRoot,
getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner,
getFocusTraversalKeys, getGraphicsConfiguration, getIconImages, getInputContext,
getListeners, getLocale, getLocationOnWindow, getModalBlocker, getModalExclusionType,
getMostRecentFocusOwner, getOwnedWindows_NoClientCode, getOwnedWindows,
getOwner_NoClientCode, getOwner, getOwnerlessWindows, getTemporaryLostComponent,
getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners,
getWindows, getWindows, getWindowStateListeners, hide, init, initDeserializedWindow,
initIDs, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isFocusableWindow,
isFocusCycleRoot, isFocused, isLocationByPlatform, isModalBlocked, isModalExcluded,
isRecursivelyVisible, isShowing, mixOnReshaping, ownedInit, pack, postEvent,
postProcessKeyEvent, postWindowEvent, preProcessKeyEvent, processEvent,
processWindowEvent, processWindowFocusEvent, processWindowStateEvent, readObject,
removeFromWindowList, removeFromWindowList, removeNotify, removeOwnedWindow,
removeWindowFocusListener, removeWindowListener, removeWindowStateListener, resetGC,
reshape, setAlwaysOnTop, setBounds, setBounds, setClientSize, setCursor,
setFocusableWindowState, setFocusCycleRoot, setIconImage, setIconImages,
setLayersOpaque, setLocationByPlatform, setLocationRelativeTo, setMinimumSize,
setModalBlocked, setModalExclusionType, setSize, setSize, setTemporaryLostComponent,
setVisible, setWarningString, show, toBack_NoClientCode, toBack,
toFront_NoClientCode, toFront, updateChildFocusableWindowState,
updateChildrenBlocking, updateWindow, writeObject

```

**Methods inherited from class** java.awt.Container

```

add, add, add, add, add, addContainerListener, addDelicately, addImpl, addNotify,
addPropertyChangeListener, addPropertyChangeListener, adjustDecendantsOnParent,
adjustDescendants, adjustListeningChildren, applyComponentOrientation,
areFocusTraversalKeysSet, canContainFocusOwner, checkAdding, checkAddToSelf, checkGD,
checkNotAWindow, clearCurrentFocusCycleRootOnHide, clearMostRecentFocusOwnerOnHide,
containsFocus, countComponents, countHierarchyMembers, createChildHierarchyEvents,
createHierarchyEvents, decreaseComponentCount, deliverEvent, dispatchEventImpl,
dispatchEventToSelf, doLayout, eventEnabled, findComponentAt, findComponentAt,
findComponentAt, findComponentAtImpl, findTraversalRoot, getAccessibleAt,
getAccessibleChild, getAccessibleChildrenCount, getAlignmentX, getAlignmentY,
getBottommostComponentIndex, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents_NoClientCode, getComponents, getComponentZOrder,
getContainerListeners, getDropTargetEventTarget, getFocusTraversalKeys,
getFocusTraversalPolicy, getHeavyweightContainer, getInsets, getLayout, getListeners,
getListenersCount, getMaximumSize, getMinimumSize, getMouseEventTarget,
getMouseEventTarget, getMouseEventTargetImpl, getMousePosition, getOpaqueShape,
getPreferredSize, getTopmostComponentIndex, hasHeavyweightDescendants,
hasLightweightDescendants, increaseComponentCount, initializeFocusTraversalKeys,
initIDs, insets, invalidate, invalidateTree, isAncestorOf, isFocusCycleRoot,
isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet,
isParentOf, isRemoveNotifyNeeded, isSameOrAncestorOf, layout, lightweightPaint,
lightweightPrint, list, list, locate, minimumSize, mixOnShowing, mixOnValidating,
mixOnZOrderChanging, nextFocusHelper, numListening, paint, paintComponents,
paintHeavyweightComponents, paramString, postProcessKeyEvent, postsOldMouseEvents,
preferredSize, preProcessKeyEvent, print, printComponents,
printHeavyweightComponents, processContainerEvent, processEvent, proxyEnableEvents,
readObject, recursiveApplyCurrentShape, recursiveApplyCurrentShape,
recursiveApplyCurrentShape, recursiveSubtractAndApplyShape,
recursiveSubtractAndApplyShape, recursiveSubtractAndApplyShape, remove, remove,
removeAll, removeContainerListener, removeDelicately, removeNotify, reparentChild,
reparentTraverse, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout,
startLWModal, stopLWModal, transferFocusBackward, transferFocusDownCycle, update,
validate, validateTree, writeObject

```

**Methods inherited from class** java.awt.Component



removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, repaintParentIfNeeded, requestFocus, requestFocus, requestFocus, requestFocus, requestFocusHelper, requestFocusHelper, requestFocusInWindow, requestFocusInWindow, requestFocusInWindow, requestFocusInWindow, resetGC, reshape, reshapeNativePeer, resize, resize, setBackground, setBounds, setBounds, setBoundsOp, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys\_NoIDCheck, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setGCFromPeer, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setRequestFocusController, setSize, setSize, setVisible, show, show, size, subtractAndApplyShape, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, update, updateCursorImmediately, validate, writeObject

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.awt.image.ImageObserver

imageUpdate

#### Methods inherited from interface java.awt.MenuContainer

getFont, postEvent, remove

#### Methods inherited from interface javax.accessibility.Accessible

getAccessibleContext

#### Methods inherited from interface java.awt.MenuContainer

getFont, postEvent, remove

#### Methods inherited from interface javax.accessibility.Accessible

getAccessibleContext

#### Methods inherited from interface javax.swing.RootPaneContainer

getContentPane, getGlassPane, getLayeredPane, getRootPane, setContentPane, setGlassPane, setLayeredPane

#### Methods inherited from interface javax.swing.TransferHandler.HasGetTransferHandler

getTransferHandler

#### Methods inherited from interface java.util.Observer

update

## Fields

(continued from last page)

## A

private Main.App **A**

Referència a l'App

---

## M

private Dades.Model **M**

Referència al model de dades

---

## MB

private javax.swing.JMenuBar **MB**

Element gràfic de la GUI: barra de menú

---

## BG

private javax.swing.ButtonGroup **BG**

Element gràfic de la GUI: agrupació de botons

---

## MenuArxiu

private javax.swing.JMenu **MenuArxiu**

Element gràfic de la GUI: menu arxiu

---

## MenuIdioma

private javax.swing.JMenu **MenuIdioma**

Element gràfic de la GUI: menu idioma

---

## MenuAjuda

private javax.swing.JMenu **MenuAjuda**

Element gràfic de la GUI: menu ajuda

---

## MenuItemNouPacient

private javax.swing.JMenuItem **MenuItemNouPacient**

Element gràfic de la GUI: menu ítem nou pacient

---

## MenuItemModificarPacient

private javax.swing.JMenuItem **MenuItemModificarPacient**

Element gràfic de la GUI: menu ítem modificar pacient

---

## MenuItemEliminarPacient

private javax.swing.JMenuItem **MenuItemEliminarPacient**

---



(continued from last page)

Element gràfic de la GUI: menu ítem eliminar pacient

---

## MenuItemObtenirHistorial

```
private javax.swing.JMenuItem MenuItemObtenirHistorial
```

Element gràfic de la GUI: menu ítem obtenir historial

---

## MenuItemGenerarInformePDF

```
private javax.swing.JMenuItem MenuItemGenerarInformePDF
```

Element gràfic de la GUI: menu ítem generar informe PDF

---

## MenuItemSortir

```
private javax.swing.JMenuItem MenuItemSortir
```

Element gràfic de la GUI: menu ítem sortir

---

## MenuItemMostrarAjuda

```
private javax.swing.JMenuItem MenuItemMostrarAjuda
```

Element gràfic de la GUI: menu ítem mostrar ajuda

---

## MenuItemIdiomaCatala

```
private javax.swing.JRadioButtonMenuItem MenuItemIdiomaCatala
```

Element gràfic de la GUI: menu ítem idioma català

---

## MenuItemIdiomaCastella

```
private javax.swing.JRadioButtonMenuItem MenuItemIdiomaCastella
```

Element gràfic de la GUI: menu ítem idioma castellà

---

## MenuItemIdiomaAngles

```
private javax.swing.JRadioButtonMenuItem MenuItemIdiomaAngles
```

Element gràfic de la GUI: menu ítem idioma anglès

---

## DialogPacient

```
private javax.swing.JDialog DialogPacient
```

Element gràfic de la GUI:

---

## Grafic1

```
private javax.swing.JLabel Grafic1
```

---

(continued from last page)

---

## LabelCodiHistorial

```
private javax.swing.JLabel LabelCodiHistorial
```

---

## LabelCodiHistorialAns

```
private javax.swing.JLabel LabelCodiHistorialAns
```

---

## LabelDarreraTransferencia

```
private javax.swing.JLabel LabelDarreraTransferencia
```

---

## LabelDarreraTransferenciaAns

```
private javax.swing.JLabel LabelDarreraTransferenciaAns
```

---

## LabelDataAlta

```
private javax.swing.JLabel LabelDataAlta
```

---

## LabelDataAltaAns

```
private javax.swing.JLabel LabelDataAltaAns
```

---

## LabelEdat

```
private javax.swing.JLabel LabelEdat
```

---

## LabelEdatAns

```
private javax.swing.JLabel LabelEdatAns
```

---

## LabelSeleccionarGrafic

```
private javax.swing.JLabel LabelSeleccionarGrafic
```

---

## LabelSeleccionarPacient

```
private javax.swing.JLabel LabelSeleccionarPacient
```

---

(continued from last page)

---

## LabelSexe

```
private javax.swing.JLabel LabelSexe
```

---

## LabelSexeAns

```
private javax.swing.JLabel LabelSexeAns
```

---

## SelectorGrafic

```
private java.awt.Choice SelectorGrafic
```

---

## SelectorPacient

```
private java.awt.Choice SelectorPacient
```

---

## TaulaAlarmes

```
private javax.swing.JTable TaulaAlarmes
```

---

## jPanel1

```
private javax.swing.JPanel jPanel1
```

---

## jScrollPane1

```
private javax.swing.JScrollPane jScrollPane1
```

---

## jSeparator1

```
private javax.swing.JSeparator jSeparator1
```

---

## Constructors

### GUI

```
public GUI(App A,  
          Model M)
```

Crea una nova instància de GUI

---

(continued from last page)

**Parameters:**

A - Referència a l'aplicació

M - Referència al Model de dades

## Methods

### initComponents

```
private void initComponents()
```

This method is called from within the constructor to initialize the form. WARNING: Do NOT modify this code. The content of this method is always regenerated by the Form Editor.

### SelectorGraficListener

```
private void SelectorGraficListener(java.awt.event.ItemEvent evt)
```

### SelectorPacientListener

```
private void SelectorPacientListener(java.awt.event.ItemEvent evt)
```

### crearMenu

```
private void crearMenu()
```

Crea el menú de l'aplicació

### mostrarDialogPacient

```
private void mostrarDialogPacient(int accio,  
    java.lang.String titol)
```

Mostra la finestra de creació/modificació d'un pacient

**Parameters:**

accio - Accio a realitzar sobre el dialog (crear/modificar)

titol - Títol del dialog

### mostrarConfirmacio

```
private int mostrarConfirmacio(java.lang.String textConfirmacio)
```

Mostra un missatge de confirmació per pantalla

**Parameters:**

textConfirmacio - Text del missatge de confirmació

### seleccionaPath

```
private java.lang.String seleccionaPath(java.lang.String nomfitxer)
```

Mostra una finestra selectora de ruta i nom de fitxer

(continued from last page)

**Parameters:**

`nomfitxer` - Nom del fitxer per defecte

---

**actualitzaParametres**

```
private void actualitzaParametres()
```

Actualitza els paràmetres de la GUI

---

**actualitzaTaula**

```
private void actualitzaTaula()
```

Actualitza la taula d'alarmes de la GUI

---

**actualitzaGrafic**

```
private void actualitzaGrafic()
```

Actualitza el gràfic d'alarmes de la GUI

---

**actualitzaDadesPacient**

```
private void actualitzaDadesPacient()
```

Actualitza les dades del pacient seleccionat a la GUI

---

**mostrarAlerta**

```
public void mostrarAlerta(java.lang.String text,  
                           int tipus)
```

Mostra un missatge d'alerta d'un tipus determinat (informació, error...)

**Parameters:**

`text` - Missatge d'alerta

---

**update**

```
public void update(java.util.Observable o,  
                   java.lang.Object arg)
```

Actualitza la vista amb les noves dades del Model (patró MVC)

**Parameters:**

`o` - Observable que notifica el canvi en el Model  
`arg` - Arguments

---

# Vista

## Class SplashScreen

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- java.awt.Window
        |-- javax.swing.JWindow
          |-- Vista.SplashScreen

```

### All Implemented Interfaces:

java.lang.Runnable, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.RootPaneContainer, javax.accessibility.Accessible

### public class **SplashScreen**

extends javax.swing.JWindow

implements javax.accessibility.Accessible, javax.swing.RootPaneContainer,

javax.swing.TransferHandler.HasGetTransferHandler, javax.accessibility.Accessible,

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.lang.Runnable

Classe que defineix la finestra Splash d'inici de l'aplicació

#### Author:

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

#### Version:

1.0 (Juny de 2010)

## Field Summary

private [V](#)

### Fields inherited from class javax.swing.JWindow

accessibleContext, rootPane, rootPaneCheckingEnabled, transferHandler

### Fields inherited from class java.awt.Window

allWindows, alwaysOnTop, anchor, base, beforeFirstShow, beforeFirstWindowShown, focusableWindowState, focusMgr, icons, inputContext, inputContextLock, isInShow, isTrayIconWindow, locationByPlatform, locationByPlatformProp, log, modalBlocker, modalExclusionType, nameCounter, opacity, opaque, OPENED, ownedWindowList, securityWarningAlignmentX, securityWarningAlignmentY, securityWarningHeight, securityWarningPointX, securityWarningPointY, securityWarningWidth, serialVersionUID, shape, showWithParent, state, syncLWRequests, systemSyncLWRequests, temporaryLostComponent, TRANSPARENT\_BACKGROUND\_COLOR, warningString, weakThis, windowFocusListener, windowListener, windowSerializedDataVersion, windowStateListener

### Fields inherited from class java.awt.Container

component, containerListener, containerSerializedDataVersion, dbg, descendantsCount, dispatcher, EMPTY\_ARRAY, focusCycleRoot, focusTraversalPolicy, focusTraversalPolicyProvider, INCLUDE\_SELF, layoutMgr, listeningBoundsChildren, listeningChildren, mixingLog, modalAppContext, modalComp, numOfHWComponents, numOfLWComponents, preserveBackgroundColor, printing, printingThreads, SEARCH\_HEAVYWEIGHTS, serialPersistentFields, serialVersionUID

#### Fields inherited from class `java.awt.Component`

accessibleContext, actionListenerK, adjustmentListenerK, appContext, background, backgroundEraseDisabled, BOTTOM\_ALIGNMENT, boundsOp, bufferStrategy, CENTER\_ALIGNMENT, changeSupport, coalesceEventsParams, coalesceMap, coalescingEnabled, componentListener, componentListenerK, componentOrientation, componentSerializedDataVersion, compoundShape, containerListenerK, cursor, dbg, dropTarget, enabled, eventCache, eventMask, FOCUS\_TRAVERSABLE\_DEFAULT, FOCUS\_TRAVERSABLE\_SET, FOCUS\_TRAVERSABLE\_UNKNOWN, focusable, focusListener, focusListenerK, focusLog, focusTraversalKeyPropertyNames, focusTraversalKeys, focusTraversalKeysEnabled, font, foreground, graphicsConfig, height, hierarchyBoundsListener, hierarchyBoundsListenerK, hierarchyListener, hierarchyListenerK, ignoreRepaint, incRate, inputMethodListener, inputMethodListenerK, isAddNotifyComplete, isFocusTraversableOverridden, isInc, isOpaqueForMixing, isPacked, itemListenerK, keyListener, keyListenerK, LEFT\_ALIGNMENT, locale, LOCK, log, maxSize, maxSizeSet, minSize, minSizeSet, mixingLog, mouseListener, mouseListenerK, mouseMotionListener, mouseMotionListenerK, mouseWheelListener, mouseWheelListenerK, name, nameExplicitlySet, nativeInLightFixer, newEventsOnly, objectLock, opaquePropertyChangeListener, ownedWindowK, parent, peer, peerFont, popups, prefSize, prefSizeSet, privateKey, requestFocusController, RIGHT\_ALIGNMENT, serialVersionUID, textListenerK, TOP\_ALIGNMENT, valid, visible, width, windowClosingException, windowFocusListenerK, windowListenerK, windowStateListenerK, x, y

#### Fields inherited from interface `java.awt.image.ImageObserver`

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

## Constructor Summary

public	<a href="#">SplashScreen</a> ( <a href="#">GUI</a> V) Crea una instància de SplashScreen
--------	---

## Method Summary

void	<a href="#">run</a> () Codi principal del thread de SplashScreen
------	---

#### Methods inherited from class `javax.swing.JWindow`

addImpl, createRootPane, getAccessibleContext, getContentPane, getGlassPane, getGraphics, getLayeredPane, getRootPane, getTransferHandler, isRootPaneCheckingEnabled, paramString, remove, repaint, setContentPane, setGlassPane, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update, windowInit

#### Methods inherited from class `java.awt.Window`

```

addNotify, addOwnedWindow, addPropertyChangeListener, addPropertyChangeListener,
addToWindowList, addWindowFocusListener, addWindowListener, addWindowStateListener,
adjustDecendantsOnParent, adjustListeningChildrenOnParent, applyCompoundShape,
applyCurrentShape, applyOpacity, applyOpaque, applyResourceBundle,
applyResourceBundle, applyShape, calculateSecurityWarningPosition,
canContainFocusOwner, clearMostRecentFocusOwnerOnHide, connectOwnedWindow,
constructComponentName, createBufferStrategy, createBufferStrategy,
deliverMouseWheelToAncestor, deserializeResources, dispatchEventImpl,
dispatchMouseWheelToAncestor, dispose, disposeImpl, doDispose, doesClassImplement,
doesImplement, eventEnabled, getAccessibleContext, getAllUnblockedWindows,
getAllWindows, getBufferStrategy, getContainer, getDocumentRoot,
getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner,
getFocusTraversalKeys, getGraphicsConfiguration, getIconImages, getInputContext,
getListeners, getLocale, getLocationOnWindow, getModalBlocker, getModalExclusionType,
getMostRecentFocusOwner, getOwnedWindows_NoClientCode, getOwnedWindows,
getOwner_NoClientCode, getOwner, getOwnerlessWindows, getTemporaryLostComponent,
getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners,
getWindows, getWindows, getWindowStateListeners, hide, init, initDeserializedWindow,
initIDs, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isFocusableWindow,
isFocusCycleRoot, isFocused, isLocationByPlatform, isModalBlocked, isModalExcluded,
isRecursivelyVisible, isShowing, mixOnReshaping, ownedInit, pack, postEvent,
postProcessKeyEvent, postWindowEvent, preProcessKeyEvent, processEvent,
processWindowEvent, processWindowFocusEvent, processWindowStateEvent, readObject,
removeFromWindowList, removeFromWindowList, removeNotify, removeOwnedWindow,
removeWindowFocusListener, removeWindowListener, removeWindowStateListener, resetGC,
reshape, setAlwaysOnTop, setBounds, setBounds, setClientSize, setCursor,
setFocusableWindowState, setFocusCycleRoot, setIconImage, setIconImages,
setLayersOpaque, setLocationByPlatform, setLocationRelativeTo, setMinimumSize,
setModalBlocked, setModalExclusionType, setSize, setSize, setTemporaryLostComponent,
setVisible, setWarningString, show, toBack_NoClientCode, toBack,
toFront_NoClientCode, toFront, updateChildFocusableWindowState,
updateChildrenBlocking, updateWindow, writeObject

```

**Methods inherited from class** java.awt.Container



```

add, add, add, add, add, addContainerListener, addDelicately, addImpl, addNotify,
addPropertyChangeListener, addPropertyChangeListener, adjustDecendantsOnParent,
adjustDescendants, adjustListeningChildren, applyComponentOrientation,
areFocusTraversalKeysSet, canContainFocusOwner, checkAdding, checkAddToSelf, checkGD,
checkNotAWindow, clearCurrentFocusCycleRootOnHide, clearMostRecentFocusOwnerOnHide,
containsFocus, countComponents, countHierarchyMembers, createChildHierarchyEvents,
createHierarchyEvents, decreaseComponentCount, deliverEvent, dispatchEventImpl,
dispatchEventToSelf, doLayout, eventEnabled, findComponentAt, findComponentAt,
findComponentAt, findComponentAtImpl, findTraversalRoot, getAccessibleAt,
getAccessibleChild, getAccessibleChildrenCount, getAlignmentX, getAlignmentY,
getBottommostComponentIndex, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents_NoClientCode, getComponents, getComponentZOrder,
getContainerListeners, getDropTargetEventTarget, getFocusTraversalKeys,
getFocusTraversalPolicy, getHeavyweightContainer, getInsets, getLayout, getListeners,
getListenersCount, getMaximumSize, getMinimumSize, getMouseEventTarget,
getMouseEventTarget, getMouseEventTargetImpl, getMousePosition, getOpaqueShape,
getPreferredSize, getTopmostComponentIndex, hasHeavyweightDescendants,
hasLightweightDescendants, increaseComponentCount, initializeFocusTraversalKeys,
initIDs, insets, invalidate, invalidateTree, isAncestorOf, isFocusCycleRoot,
isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet,
isParentOf, isRemoveNotifyNeeded, isSameOrAncestorOf, layout, lightweightPaint,
lightweightPrint, list, list, locate, minimumSize, mixOnShowing, mixOnValidating,
mixOnZOrderChanging, nextFocusHelper, numListening, paint, paintComponents,
paintHeavyweightComponents, paramString, postProcessKeyEvent, postsOldMouseEvents,
preferredSize, preProcessKeyEvent, print, printComponents,
printHeavyweightComponents, processContainerEvent, processEvent, proxyEnableEvents,
readObject, recursiveApplyCurrentShape, recursiveApplyCurrentShape,
recursiveApplyCurrentShape, recursiveSubtractAndApplyShape,
recursiveSubtractAndApplyShape, recursiveSubtractAndApplyShape, remove, remove,
removeAll, removeContainerListener, removeDelicately, removeNotify, reparentChild,
reparentTraverse, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout,
startLWModal, stopLWModal, transferFocusBackward, transferFocusDownCycle, update,
validate, validateTree, writeObject

```

**Methods inherited from class** java.awt.Component



```
removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener,
removeKeyListener, removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removeNotify, removePropertyChangeListener,
removePropertyChangeListener, repaint, repaint, repaint, repaint,
repaintParentIfNeeded, requestFocus, requestFocus, requestFocus, requestFocus,
requestFocusHelper, requestFocusHelper, requestFocusInWindow, requestFocusInWindow,
requestFocusInWindow, requestFocusInWindow, resetGC, reshape, reshapeNativePeer,
resize, resize, setBackground, setBounds, setBounds, setBoundsOp,
setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeys_NoIDCheck, setFocusTraversalKeys, setFocusTraversalKeysEnabled,
setFont, setForeground, setGCFromPeer, setIgnoreRepaint, setLocale, setLocation,
setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize,
setRequestFocusController, setSize, setSize, setVisible, show, show, size,
subtractAndApplyShape, toString, transferFocus, transferFocusBackward,
transferFocusUpCycle, update, updateCursorImmediately, validate, writeObject
```

#### Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

#### Methods inherited from interface `java.awt.image.ImageObserver`

```
imageUpdate
```

#### Methods inherited from interface `java.awt.MenuContainer`

```
getFont, postEvent, remove
```

#### Methods inherited from interface `javax.accessibility.Accessible`

```
getAccessibleContext
```

#### Methods inherited from interface `javax.accessibility.Accessible`

```
getAccessibleContext
```

#### Methods inherited from interface `javax.swing.RootPaneContainer`

```
getContentPane, getGlassPane, getLayeredPane, getRootPane, setContentPane,
setGlassPane, setLayeredPane
```

#### Methods inherited from interface `javax.swing.TransferHandler.HasGetTransferHandler`

```
getTransferHandler
```

#### Methods inherited from interface `java.lang.Runnable`

```
run
```

## Fields

(continued from last page)

## V

```
private Vista.GUI v
```

## Constructors

### SplashScreen

```
public SplashScreen(GUI v)
```

Crea una instància de SplashScreen

**Parameters:**

v - Referència a la GUI principal

## Methods

### run

```
public void run()
```

Codi principal del thread de SplashScreen

# Vista

## Interface Vista

All Known Implementing Classes:

[DialogPacient](#), [GUI](#)

---

public interface **Vista**  
extends java.util.Observer

Interfície d'una vista per a aplicar el patró MVC (Model-View-Controller)

**Author:**

Oriol Ciurana (oriol.ciurana@entel.upc.edu)

**Version:**

1.0 (Juny de 2010)

---

Methods inherited from interface <code>java.util.Observer</code>
update

---

## D.3 Codi de l'aplicació

Codi D.1: (AppSeguiment) Bluetooth.java

```
1 package Bluetooth;
2
3 import Constants.LANG;
4 import Dades.Historial;
5 import Dades.Model;
6 import Dades.Pacient;
7 import Utils.Utils;
8 import Vista.GUI;
9 import java.io.IOException;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 import javax.bluetooth.*;
13 import javax.microedition.io.*;
14 import javax.swing.JOptionPane;
15
16 /**
17  * Classe que obté l'historial d'alarmes d'un pacient mitjançant Bluetooth
18  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
19  * @version 1.0 (Juny de 2010)
20  */
21 public class Bluetooth extends Thread implements DiscoveryListener {
22
23     /** Referència a la GUI on mostrar les alertes */
24     private GUI V;
25     /** Referència al Model */
26     private Model M;
27     /** Pacient del que es vol obtenir l'historial */
28     private Pacient P;
29     /** UUID del servei a buscar mitjançant el SDP */
30     protected UUID Servei;
31     /** Objecte de connexió */
32     private Connexio C;
33     /** Dispositiu local */
34     private LocalDevice LD;
35     /** Agent de descobriment */
36     private DiscoveryAgent DA;
37     /** URL de connexió */
38     private String url;
39     /** Indica si s'ha trobat un servei PPStop o no */
40     private boolean serveiPPStop;
41     /** Identifica el final de la transferència d'historial */
42     private boolean doing;
43
44     /**
45      * Crea una instància de Bluetooth
46      * @param M GUI on mostrar les alertes corresponents
47      * @param M Model de dades
48      * @param P Pacient del que es vol obtenir l'historial
49      */
50     public Bluetooth(GUI V, Model M, Pacient P){
51         this.V = V;
52         this.M = M;
53         this.P = P;
54         Servei = new UUID(0x1101);
55         C = new Connexio();
56         try {
57             LD = LocalDevice.getLocalDevice();
58             DA = LD.getDiscoveryAgent();
59             V.mostrarAlerta(LANG.TXT_ALERTA_OPERACIOPOTTRIGAR[M.getIdioma()],JOptionPane
                .INFORMATION_MESSAGE);
60         } catch (BluetoothStateException ex) {
61             V.mostrarAlerta(LANG.TXT_ERROR_NOBLUETOOTH[M.getIdioma()],JOptionPane.
                ERROR_MESSAGE);
62         }
63     }
```

```

64
65 /**
66  * Mètode que és cridat quan s'ha descobert un nou dispositiu a través del SDP
67  * @param btDevice Dispositiu remot descobert
68  * @param cod Codi identificador de la classe de dispositiu descobert
69  */
70 public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {}
71
72 /**
73  * Mètode que és cridat quan s'ha completat la búsqueda de dispositius (DDP)
74  * @param discType Identificador del motiu de la finalització DDP
75  */
76 public void inquiryCompleted(int discType) {}
77
78 /**
79  * Inicia una búsqueda de serveis (SDP) en el dispositiu especificat
80  * @param device Dispositiu sobre el que iniciar el SDP
81  */
82 private void startServiceSearch(RemoteDevice device) {
83     System.out.println("Iniciant búsqueda de serveis");
84     serveiPPStop = false;
85     try {
86         UUID uuids[] = new UUID[] { Servei };
87         DA.searchServices(null, uuids, device, this);
88     } catch (Exception e) {}
89 }
90
91 /**
92  * Mètode que és cridat quan s'ha completat la búsqueda de serveis (SDP)
93  * @param transID Identificador del recurs que ha iniciat el SDP
94  * @param respCode Codi identificador de l'estat de la transacció
95  */
96 public void serviceSearchCompleted(int transID, int respCode) {
97     System.out.println("Búsqueda de serveis finalitzada");
98     if (!serveiPPStop)
99         V.mostrarAlerta(LANG.TXT_ALERTA_OPERACIOPOTTRIGAR[M.getIdioma()], JOptionPane
100             .INFORMATION_MESSAGE);
101 }
102
103 /**
104  * Mètode que és cridat quan s'ha completat la búsqueda de serveis (SDP)
105  * @param transID Identificador del recurs que ha apuntat el resultat
106  * @param respCode Llista de serveis trobats durant la búsqueda de serveis (SDP)
107  */
108 public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
109     serveiPPStop = true;
110     url = servRecord[0].getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT,
111         false);
112     System.out.println("Servei PPStop trobat: " + url);
113     DA.cancelServiceSearch(transID);
114     handleConnection(url);
115 }
116
117 /**
118  * Realitza la connexió amb el dispositiu remot i n'obté l'historial d'alarmes
119  * @param url Adreça de connexió
120  */
121 private void handleConnection(final String url) {
122     int numAlarmes = 0;
123     try {
124         StreamConnection SC = (StreamConnection) Connector.open(url);
125         C.obrirDataStreams(SC);
126         C.enviarMSG("T");
127         Historial H = C.rebreHIST();
128         if (H.getTZid()==null)
129             V.mostrarAlerta(LANG.TXT_ERROR_TRANSFERENCIAHISTORIAL[M.getIdioma()],
130                 JOptionPane.ERROR_MESSAGE);
131         numAlarmes = M.setHistorial(P, H);
132         C.tancarDataStreams();
133         SC.close();
134     }
135 }

```

```

131     } catch (IOException ex) {
132         V.mostrarAlerta(LANG.TXT_ERROR_TRANSFERENCIAHISTORIAL[M.getIdioma()],
133             JOptionPane.ERROR_MESSAGE);
134         Logger.getLogger(Bluetooth.class.getName()).log(Level.SEVERE, null, ex);
135     } finally {
136         V.mostrarAlerta(Utils.substituirStr(LANG.TXT_ALERTA_HISTORIALREBUT[M.
137             getIdioma()], "{X}", String.valueOf(numAlarmes)), JOptionPane.
138             INFORMATION_MESSAGE);
139         doing = false;
140     }
141 }
142
143 @SuppressWarnings("empty-statement")
144 @Override
145 public void run(){
146     doing = true;
147     startServiceSearch(new RemoteDevice(P.getCodiHistorial()){});
148     while(doin);
149 }
150 }

```

## Codi D.2: (AppSeguiment) Connexio.java

```

1 package Bluetooth;
2
3
4 import Dades.Historial;
5 import java.io.DataInputStream;
6 import java.io.DataOutputStream;
7 import java.io.IOException;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10 import javax.microedition.io.StreamConnection;
11
12 /**
13  * Classe que obre els streams d'e I/O per a la transmissió i recepció de dades
14  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
15  * @version 1.0 (Juny de 2010)
16  */
17 public class Connexio {
18
19     /** Stream d'entrada de dades */
20     private DataInputStream dis;
21     /** Stream de sortida de dades */
22     private DataOutputStream dos;
23
24     /**
25      * Obtené els Streams de I/O a partir d'un l'StreamConnection
26      * @param SC      Stream de la connexió
27      */
28     public void obrirDataStreams(StreamConnection SC){
29         try {
30             dis = SC.openDataInputStream();
31             dos = SC.openDataOutputStream();
32         } catch (IOException ex) {
33             ex.printStackTrace();
34         }
35     }
36
37     /**
38      * Tanca els Streams de I/O
39      */
40     public void tancarDataStreams(){
41         try {
42             dis.close();
43             dos.close();
44         } catch (IOException ex) {
45             ex.printStackTrace();
46         }
47     }
48 }

```



```

46     }
47 }
48
49 /**
50  * Transmet un Missatge a través de l'OutputStream
51  * @param msg    Missatge a enviar
52  */
53 public void enviarMSG(String msg){
54     try {
55         dos.write(msg.getBytes());
56         dos.flush();
57     } catch (IOException ex) {
58         ex.printStackTrace();
59     }
60 }
61
62 /**
63  * Rep un Historial a través de l'OutputStream
64  * @return Historial rebut
65  */
66 public Historial rebreHIST(){
67     Historial H = new Historial();
68     try {
69         byte[] b = new byte[10000];
70         dis.read(b);
71         H.fromByteStream(b);
72     } catch (IOException ex) {
73         Logger.getLogger(Connexio.class.getName()).log(Level.SEVERE, null, ex);
74     }
75     return H;
76 }
77
78 }

```

### Codi D.3: (AppSeguiment) CONST.java

```

1
2 package Constants;
3
4 /**
5  * Interfície que conté totes les constants numèriques utilitzades per l'aplicació
6  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
7  * @version 1.0 (Juny de 2010)
8  */
9 public interface CONST {
10
11     /** L'idioma de la interfície de l'aplicació és el Català */
12     public static final int IDIOMA_CA = 0;
13     /** L'idioma de la interfície de l'aplicació és el Castellà */
14     public static final int IDIOMA_ES = 1;
15     /** L'idioma de la interfície de l'aplicació és l'Anglès */
16     public static final int IDIOMA_EN = 2;
17
18     /** El tipus de data és la data sencera */
19     public static final int DATA_SENCERA = 0;
20     /** El tipus de data és la data curta */
21     public static final int DATA_CURTA = 1;
22     /** El tipus de data és la data mensual */
23     public static final int DATA_MENSUAL = 2;
24     /** El tipus de data és l'any */
25     public static final int DATA_ANY = 3;
26     /** El tipus de data és el mes */
27     public static final int DATA_MES = 4;
28     /** El tipus de data és el mes en format text */
29     public static final int DATA_MESTXT = 5;
30     /** El tipus de data és el dia */
31     public static final int DATA_DIA = 6;
32     /** El tipus de data és l'hora */
33     public static final int DATA_HORA = 7;

```

```

34
35     /** El tipus d'acció és la de crear un nou pacient */
36     public static final int PACIENT_CREAR = 0;
37     /** El tipus d'acció és la de modificar el pacient que s'està tractant */
38     public static final int PACIENT_MODIFICAR = 1;
39
40     /** El tipus de gràfic a mostrar és el de l'evolució d'alarmes mensual */
41     public static final int GRAFIC_EVOLUCIOMENSUAL = 0;
42     /** El tipus de gràfic a mostrar és el de l'evolució d'alarmes dels darrers 30 dies
43         */
44     public static final int GRAFIC_DARRERS30DIES = 1;
45     /** El tipus de gràfic a mostrar és el de la distribució horària de les alarmes */
46     public static final int GRAFIC_DISTRIBUCIOHORARIA = 2;
47     /** El tipus de gràfic a mostrar és el de la distribució diurna/nocturna de les
48         alarmes */
49     public static final int GRAFIC_DISTRIBUCIODIURNANOCTURNA = 3;
50 }

```

#### Codi D.4: (AppSeguiment) LANG.java

```

1
2 package Constants;
3
4 /**
5  * Interfície que conté totes les constants d'idioma utilitzades per l'aplicació
6  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
7  * @version 1.0 (Juny de 2010)
8  */
9 public interface LANG {
10
11     /** Label del títol de l'informe de seguiment */
12     public static final String[] TXT_INFORME = {"Informe de seguiment", "Informe de
13         seguimiento", "Monitoring report"};
14     /** Label de data de generació */
15     public static final String[] TXT_DATAGENERACIO = {"Data de generació", "Fecha de
16         generación", "Generation date"};
17     /** Label de dades del pacient */
18     public static final String[] TXT_DADESPACIENT = {"Dades del pacient", "Datos del
19         paciente", "Patient information"};
20     /** Label de codi d'historial */
21     public static final String[] TXT_CODIHISTORIAL = {"Codi d'historial", "Código de
22         historial", "History code"};
23     /** Label de nom */
24     public static final String[] TXT_NOM = {"Nom", "Nombre", "Name"};
25     /** Label de cognoms */
26     public static final String[] TXT_COGNOMS = {"Cognoms", "Apellidos", "Surnames"};
27     /** Label d'edat */
28     public static final String[] TXT_EDAT = {"Edat", "Edad", "Age"};
29     /** Label de data de naixement */
30     public static final String[] TXT_DATANAIXEMENT = {"Data de naixement", "Fecha de
31         nacimiento", "Date of birth"};
32     /** Label de sexe */
33     public static final String[] TXT_SEXE = {"Sexe", "Sexo", "Genre"};
34     /** Label de tipus de sexe */
35     public static final String[][] TXT_SEXETXT = {{ "Masculí", "Femení", "Masculino", "
36         Femenino"}, {"Male", "Female"} };
37     /** Label de data d'alta */
38     public static final String[] TXT_DATAALTA = {"Data d'alta", "Fecha de alta", "Date
39         added"};
40     /** Label de darrera transferència */
41     public static final String[] TXT_DARRERATRANSFERENCIA = {"Darrera transferència", "
42         Última transferencia", "Last data transfer"};
43     /** Label de gràfiques estadístiques */
44     public static final String[] TXT_GRAFIQUESESTADISTQUES = {"Gràfiques estadístiques",
45         "Gráficas estadísticas", "Statistical graphics"};
46     /** Label d'historial d'alarmes */
47     public static final String[] TXT_HISTORIALALARMES = {"Historial d'alarmes", "
48         Historial de alarmas", "Alarm history"};
49
50 }

```

```

39  /** Label d'any */
40  public static final String[] TXT_ANY = {"Any", "Año", "Year"};
41  /** Label d'anys */
42  public static final String[] TXT_ANYS = {"anys", "años", "years"};
43  /** Label de mes */
44  public static final String[] TXT_MES = {"Mes", "Mes", "Month"};
45  /** Label de tipus de mes */
46  public static final String[][] TXT_MESTXT = {
47      {"Gener", "Febrer", "Març", "Abril", "Maig", "Juny", "Juliol", "Agost", "Setembre", "
48      Octubre", "Novembre", "Deseembre"},
49      {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre",
50      "Octubre", "Noviembre", "Diciembre"},
51      {"January", "February", "March", "April", "May", "June", "July", "August", "September", "
52      October", "November", "December"}
53  };
54  /** Label de dia */
55  public static final String[] TXT_DIA = {"Dia", "Día", "Day"};
56  /** Label d'hora */
57  public static final String[] TXT_HORA = {"Hora", "Hora", "Hour"};
58  /** Label d'alarmes diürnes */
59  public static final String[] TXT_DIURNES = {"Diürnes", "Diurnas", "Diurnal"};
60  /** Label d'alarmes nocturnes */
61  public static final String[] TXT_NOCTURNES = {"Nocturnes", "Nocturnas", "Nocturnal"};
62  /** Label d'indicador de pàgina */
63  public static final String[] TXT_PAGINADE = {"Pàgina {X} de ", "Página {X} de ", "Page
64  {X} of "};
65  /** Label de la gràfica de tipus 1 */
66  public static final String[] TXT_GRAFIC1 = {"Evolució mensual del nombre d'alarmes",
67  "Evolución mensual del número de alarmas", "Monthly Alarm number evolution"};
68  /** Label de la gràfica de tipus 2 */
69  public static final String[] TXT_GRAFIC2 = {"Evolució del nombre d'alarmes dels
70  darrers 30 dies", "Evolución del numero de alarmas de los últimos 30 días", "Last
71  30 day alarm number evolution"};
72  /** Label de la gràfica de tipus 3 */
73  public static final String[] TXT_GRAFIC3 = {"Distribució horària de les alarmes (%)
74  ", "Distribución horaria de las alarmas (%)", "Hourly alarm distribution (%)"};
75  /** Label de la gràfica de tipus 4 */
76  public static final String[] TXT_GRAFIC4 = {"Distribucio (diürna/nocturna) de les
77  alarmes", "Distribución (diurna/nocturna) de las alarmas", "Diurnal/nocturnal
78  alarm distribution"};
79  /** Label de diferència de temps: < 1 minut */
80  public static final String[] TXT_MENYSUNMINUT = {"< 1 minut", "< 1 minuto", "< 1
81  minute"};
82  /** Label de diferència de temps: minut */
83  public static final String[] TXT_1MINUT = {"minut", "minuto", "minute"};
84  /** Label de diferència de temps: minuts */
85  public static final String[] TXT_XMINUTS = {"minuts", "minutos", "minutes"};
86  /** Label de diferència de temps: 1 hora */
87  public static final String[] TXT_1HORA = {"hora", "hora", "hour"};
88  /** Label de diferència de temps: hores */
89  public static final String[] TXT_XHORES = {"hores", "horas", "hours"};
90  /** Label de diferència de temps: dia */
91  public static final String[] TXT_IDIA = {"dia", "día", "day"};
92  /** Label de diferència de temps: dies */
93  public static final String[] TXT_XDIES = {"dies", "días", "days"};
94  /** Label del menú Arxiu */
95  public static final String[] MENU_ARXIU = {"Arxiu", "Archivo", "File"};
96  /** Label del menú Idioma */
97  public static final String[] MENU_IDIOMA = {"Idioma", "Idioma", "Language"};
98  /** Label del menú Ajuda */
99  public static final String[] MENU_AJUDA = {"Ajuda", "Ayuda", "Help"};
100  /** Label del menú Arxiu: Nou Pacient */
101  public static final String[] MENU_NOUPACIENT = {"Nou pacient", "Nuevo paciente", "New
102  patient"};
103  /** Label del menú Arxiu: Modificar Pacient */
104  public static final String[] MENU_MODIFICARPACIENT = {"Modificar pacient", "Modificar
105  paciente", "Modify patient"};
106  /** Label del menú Arxiu: Modificar Pacient */
107  public static final String[] MENU_ELIMINARPACIENT = {"Eliminar pacient", "Eliminar
108  paciente", "Delete patient"};

```

```

95  /** Label del menú Arxiu: Obtenir Historial */
96  public static final String[] MENU_OBTENIRHISTORIAL = {"Obtenir historial","Obtener
    historial","Get history"};
97  /** Label del menú Arxiu: Generar Informe PDF */
98  public static final String[] MENU_GENERARINFORMEPDF = {"Generar informe PDF","
    Generar informe PDF","Generate PDF report"};
99  /** Label del menú Arxiu: Sortir */
100 public static final String[] MENU_SORTIR = {"Sortir","Salir","Exit"};
101 /** Label del menú Idioma: Català */
102 public static final String[] MENU_CATALA = {"Català","Catalán","Catalan"};
103 /** Label del menú Idioma: Castellà */
104 public static final String[] MENU_CASTELLA = {"Castellà","Castellano","Spanish"};
105 /** Label del menú Idioma: Anglès */
106 public static final String[] MENU_ANGLES = {"Anglès","Inglés","English"};
107 /** Label del menú Ajuda: Mostrar Ajuda */
108 public static final String[] MENU_MOSTRARAJUDA = {"Mostrar ayuda","Mostrar ayuda","
    Show help"};
109 /** Label de selecció de pacient */
110 public static final String[] TXT_SELECCIONARPACIENT = {"Selecciona el pacient a
    analitzar:","Selecciona el paciente a analizar:","Select the patient to analyze:
    "};
111 /** Label de selecció de gràfic */
112 public static final String[] TXT_SELECCIONARGRAFIC = {"Selecciona el tipus de gràfic
    :","Selecciona el tipo de gráfico:","Please select the desired graph:"};
113 /** Labrl d'introducció de dades del nou pacient */
114 public static final String[] TXT_DIALOGPACIENT1 = {"Introduiu les dades del nou
    pacient:","Introducid las fechas del nuevo paciente:","Insert the new patient
    data:"};
115 /** Label de modificació de dades d'un pacient */
116 public static final String[] TXT_DIALOGPACIENT2 = {"Modifiqueu les dades del pacient
    :","Modifica los datos del paciente:","Modify the patient data:"};
117 /** Label d'alta de pacient */
118 public static final String[] TXT_DONARDALTA = {"Donar d'alta","Dar de alta","
    Register"};
119 /** Label de modificacio de pacient */
120 public static final String[] TXT_MODIFICARDADES = {"Modificar dades","Modificar
    datos","Modify data"};
121 /** Label de confirmació: eliminar pacient */
122 public static final String[] TXT_CONFIRMACIO_ELIMINAR = {"Segur que vols eliminar el
    pacient: {X} ?","¿Seguro que quieres eliminar el paciente: {X} ?","Are you sure
    to delete the patient: {X} ?"};
123 /** Label d'error: nom buit */
124 public static final String[] TXT_ERROR_NOM = {"El camp 'nom' està buit","El campo '
    nombre' está vacío","The 'name' field is empty"};
125 /** Label d'error: cognoms buit */
126 public static final String[] TXT_ERROR_COGNOMS = {"El camp 'cognoms' està buit","El
    campo 'apellidos' está vacío","The 'surnames' field is empty"};
127 /** Label d'error: data de naixement no vàlida */
128 public static final String[] TXT_ERROR_DATANAIXEMENT = {"El camp 'data de naixement'
    no té un format vàlid","El campo 'fecha de nacimiento' no tiene un formato
    válido","The field 'birth date' has no valid format"};
129 /** Label d'error: codi d'historial no vàlid */
130 public static final String[] TXT_ERROR_CODIHISTORIAL = {"El camp 'codi historial' no
    té un format vàlid","El campo 'código historial' no tiene un formato válido","
    The 'history code' field has no valid format"};
131 /** Label d'error: codi d'historial ja existent */
132 public static final String[] TXT_ERROR_CODIHISTORIALEXISTEIX = {"El camp 'codi
    historial' conté un codi ja existent en el sistema","El campo 'código historial'
    contiene un código ya existente en el sistema","The 'history code' field has a
    code thas is already in the system"};
133 /** Label d'error: no es pot generar l'informe perquè no hi ha alarmes */
134 public static final String[] TXT_ERROR_NOALARMES = {"No es pot generar l'informe
    porque no hi ha alarmes enmagatzemades","No se puede generar el informe porque
    no hay alarmas almacenadas","The report can not be generated because there are
    no stored alarms"};
135 /** Label d'error: no es pot generar l'informe perquè no hi ha alarmes */
136 public static final String[] TXT_ERROR_GENERACIOINFORME = {"Hi ha hagut un error
    durant la generació de l'informe PDF","Ha ocurrido un error durante la
    generacion del informe PDF","An error has occurred while generating the PDF
    report"};

```

```

137  /** Label d'error: no hi ha bluetooth */
138  public static final String[] TXT_ERROR_NOBLUETOOTH = {"No es pot obtenir l'historial
    perquè no s'ha trobat cap dispositiu Bluetooth a la vostra màquina", "No se
    puede obtener el historial porque no se ha encontrado ningún dispositivo
    Bluetooth en vuestra máquina", "The history can not be obtained because no
    Blueooth device has found in your machine"};
139  /** Label d'error: no s'ha trobat servei PPStop */
140  public static final String[] TXT_ERROR_NOSERVEIPPSTOP = {"No s'ha trobat cap servei
    PPStop associat a aquest pacient", "No se ha encontrado ningún servicio PPStop
    asociado a este paciente", "No PPStop service that matches this patient has been
    found"};
141  /** Label d'error: error en la transferència d'historial */
142  public static final String[] TXT_ERROR_TRANSFERENCIAHISTORIAL = {"Hi ha hagut un
    error durant la transferència de l'historial, si us plau intenteu-ho de nou", "Ha
    habido un error durante la transferencia del historial, por favor intentalo de
    nuevo", "There has been an error during the history transfer process, please try
    again"};
143  /** Label d'error: no hi ha cap pacient seleccionat */
144  public static final String[] TXT_ERROR_NOPACIENT = {"No hi ha cap pacient
    seleccionat", "No hay ningún paciente seleccionado", "There is no selected patient
    "};
145  /** Label d'alerta: l'operació pot trigar uns segons */
146  public static final String[] TXT_ALERTA_OPERACIOPOTTRIGAR = {"Aquesta operació pot
    trigar uns segons, mantingueu-vos a l'espera...", "Esta operación puede tardar
    unos segundos, mantengase a la espera...", "This operation may take a few seconds
    , please wait..."};
147  /** Label d'alerta: historial rebut */
148  public static final String[] TXT_ALERTA_HISTORIALREBUT = {"La recepció de l'
    historial ha finalitzat: {X} alarmes rebudes", "La recepción del historial ha
    finalizado: {X} alarmas recibidas", "The history transfer has been finished: {X}
    recived alarms"};
149
150 }

```

## Codi D.5: (AppSeguiment) Alarma.java

```

1  package Dades;
2
3
4  import java.util.Date;
5
6  /**
7   * Classe representativa d'una alarma
8   * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
9   * @version 1.0 (Juny de 2010)
10  */
11  public class Alarma {
12
13      /** CodiHistorial de l'alarma */
14      private String CodiHistorial;
15
16      /** Data de l'alarma */
17      private Date DataAlarma;
18
19      /**
20       * Crea una instància d'Alarma
21       * @param Timestamp Timestamp de l'alarma
22       */
23      public Alarma(String CodiHistorial, Date DataAlarma){
24          this.CodiHistorial = CodiHistorial;
25          this.DataAlarma = DataAlarma;
26      }
27
28      /**
29       * Obte el codi de l'historial de l'alarma
30       * @return Codi de l'historial de l'alarma
31       */
32      public String getCodiHistorial(){
33          return CodiHistorial;

```

```

34     }
35
36     /**
37      * Obte la data de l'alarma
38      * @return Data de l'alarma
39      */
40     public Date getDataAlarma(){
41         return DataAlarma;
42     }
43
44     /**
45      * Representació textual de la classe Alarma, per a funcions de debugging
46      * @return Representacio textual de la classe
47      */
48     @Override
49     public String toString(){
50         return ("CodiHistorial:"+CodiHistorial+" / DataAlarma:"+DataAlarma);
51     }
52
53 }

```

## Codi D.6: (AppSeguiment) DB.java

```

1
2 package Dades;
3
4 import Utils.Utils;
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.PreparedStatement;
8 import java.sql.ResultSet;
9 import java.sql.SQLException;
10 import java.sql.Statement;
11 import java.util.ArrayList;
12 import java.util.HashMap;
13 import java.util.List;
14 import java.util.Map;
15 import java.util.logging.Level;
16 import java.util.logging.Logger;
17
18 /**
19  * Classe que permet interactuar amb la base de dades JavaDB(derby)
20  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
21  * @version 1.0 (Juny de 2010)
22  */
23 public class DB {
24
25     /** Objecte de connexio a la base de dades */
26     private Connection C;
27
28     /**
29      * Crea una instància de DB; crea la base de dades i les taules si no existeixen
30      */
31     public DB(){
32         try {
33             Class.forName("org.apache.derby.jdbc.EmbeddedDriver");
34         } catch (ClassNotFoundException ex) {
35             Logger.getLogger(DB.class.getName()).log(Level.SEVERE, null, ex);
36         }
37         try {
38             Connectar();
39             CrearTaules();
40             Desconnectar();
41         } catch (SQLException ex) {
42             Logger.getLogger(DB.class.getName()).log(Level.SEVERE, null, ex);
43         }
44     }
45
46     /**

```

```

47     * Estableix una connexió amb la base de dades
48     */
49     public void Connectar(){
50         try {
51             C = DriverManager.getConnection("jdbc:derby:ppstop;create=true");
52         } catch (SQLException ex) {
53             Logger.getLogger(DB.class.getName()).log(Level.SEVERE, null, ex);
54         }
55     }
56
57     /**
58     * Tanca la connexio amb la base de dades
59     */
60     public void Desconnectar(){
61         try {
62             C.close();
63         } catch (SQLException ex) {
64             Logger.getLogger(DB.class.getName()).log(Level.SEVERE, null, ex);
65         }
66     }
67
68     /**
69     * Crea les taules a la base de dades si aquestes no existeixen
70     */
71     private void CrearTaules() throws SQLException{
72         Statement S = null;
73         try {
74             S = C.createStatement();
75             S.executeQuery("SELECT 1 FROM Pacients");
76         } catch (SQLException ex) {
77             S.executeUpdate("CREATE TABLE Config(" +
78                 "Idioma INTEGER)");
79             S.executeUpdate("CREATE TABLE Pacients(" +
80                 "CodiHistorial VARCHAR(12), " +
81                 "Nom VARCHAR(20), " +
82                 "Cognoms VARCHAR(50), " +
83                 "DataNaixement TIMESTAMP, " +
84                 "Sexe INTEGER, " +
85                 "DataAlta TIMESTAMP, " +
86                 "DarreraTransferencia TIMESTAMP, " +
87                 "PRIMARY KEY (CodiHistorial))");
88             S.executeUpdate("CREATE TABLE Alarmes(" +
89                 "CodiHistorial VARCHAR(12), " +
90                 "DataAlarma TIMESTAMP, " +
91                 "PRIMARY KEY (DataAlarma,CodiHistorial), " +
92                 "FOREIGN KEY (CodiHistorial) REFERENCES Pacients ON DELETE CASCADE)"
93                 );
94             S.executeUpdate("INSERT INTO Config (Idioma) VALUES (1)");
95         }
96     }
97
98     /**
99     * Selecciona l'idioma de l'aplicació
100    * @return Idioma de l'aplicacio
101    */
102    public int SelectIdioma(){
103        int idioma = 0;
104        try {
105            ResultSet RS;
106            Statement S = C.createStatement();
107            RS = S.executeQuery("SELECT Idioma FROM Config");
108            RS.next();
109            idioma = RS.getInt("Idioma");
110        } catch (SQLException ex) {}
111        return idioma;
112    }
113
114    /**
115    * Actualitza l'idioma de l'aplicació
116    * @param idioma      Nou idioma de l'aplicació

```

```

116     */
117     public void UpdateIdioma(int idioma){
118         try {
119             PreparedStatement PS = C.prepareStatement("UPDATE Config SET Idioma=?");
120             PS.setInt(1, idioma);
121             PS.executeUpdate();
122         } catch (SQLException ex) {}
123     }
124
125     /**
126     * Fa una selecció de pacients de la taula Pacients segons una condició
127     * @param condicio Condició de la selecció
128     * @return Llista de pacients seleccionats
129     */
130     public List<Pacient> SelectPacients(String condicio){
131         List<Pacient> Pacients = new ArrayList<Pacient>();
132         try {
133             ResultSet RS;
134             Statement S = C.createStatement();
135             if (condicio==null){
136                 RS = S.executeQuery("SELECT * FROM Pacients ORDER BY Cognoms ASC, Nom
137                                     ASC");
138             } else {
139                 RS = S.executeQuery("SELECT * FROM Pacients WHERE " + condicio + " ORDER
140                                     BY Cognoms ASC, Nom ASC");
141             }
142             while (RS.next())
143                 Pacients.add(new Pacient(
144                     RS.getString("CodiHistorial"),
145                     RS.getString("Nom"),
146                     RS.getString("Cognoms"),
147                     RS.getTimestamp("DataNaixement"),
148                     RS.getInt("Sexe"),
149                     RS.getTimestamp("DataAlta"),
150                     RS.getTimestamp("DarreraTransferencia")
151                 ));
152         } catch (SQLException ex) {}
153         return Pacients;
154     }
155
156     /**
157     * Insereix un pacient a la taula Pacients
158     * @param Pacient Pacient a inserir
159     */
160     public void InsertPacient(Pacient Pacient){
161         try {
162             PreparedStatement PS = C.prepareStatement("INSERT INTO Pacients (
163                 CodiHistorial,Nom,Cognoms,DataNaixement,Sexe,DataAlta,
164                 DarreraTransferencia) VALUES (?, ?, ?, ?, ?, ?, ?)");
165             PS.setString(1, Pacient.getCodiHistorial());
166             PS.setString(2, Pacient.getNom());
167             PS.setString(3, Pacient.getCognoms());
168             PS.setTimestamp(4, Utils.obtenirTimestampSQL(Pacient.getDataNaixement()));
169             PS.setInt(5, Pacient.getSexe());
170             PS.setTimestamp(6, Utils.obtenirTimestampSQL(Pacient.getDataAlta()));
171             PS.setTimestamp(7, Utils.obtenirTimestampSQL(Pacient.getDarreraTransferencia()));
172             PS.executeUpdate();
173         } catch (SQLException ex) {}
174     }
175
176     /**
177     * Actualitza un pacient de la taula Pacients
178     * @param Pacient Pacient a actualitzar
179     * @param condicio Condició de l'actualització
180     */
181     public void UpdatePacient(Pacient Pacient, String condicio){
182         try {
183             PreparedStatement PS = C.prepareStatement("UPDATE Pacients SET CodiHistorial
184                 =?,Nom=?,Cognoms=?,DataNaixement=?,Sexe=?,DataAlta=?,

```



```

180         DarreraTransferencia=? WHERE " + condicio);
181     PS.setString(1, Pacient.getCodiHistorial());
182     PS.setString(2, Pacient.getNom());
183     PS.setString(3, Pacient.getCognoms());
184     PS.setTimestamp(4, Utils.obtenirTimestampSQL(Pacient.getDataNaixement()));
185     PS.setInt(5, Pacient.getSexe());
186     PS.setTimestamp(6, Utils.obtenirTimestampSQL(Pacient.getDataAlta()));
187     PS.setTimestamp(7, Utils.obtenirTimestampSQL(Pacient.getDarreraTransferencia()));
188     PS.executeUpdate();
189     } catch (SQLException ex) {
190         Logger.getLogger(DB.class.getName()).log(Level.SEVERE, null, ex);
191     }
192 }
193 /**
194  * Elimina un pacient de la taula Pacients (i les files referenciades de la taula
195  * Alarmes)
196  * @param Pacient Pacient a eliminar
197  */
198 public void DeletePacient(String condicio){
199     try {
200         Statement S = C.createStatement();
201         S.executeUpdate("DELETE FROM Pacients WHERE " + condicio);
202     } catch (SQLException ex) {
203         Logger.getLogger(DB.class.getName()).log(Level.SEVERE, null, ex);
204     }
205 }
206 /**
207  * Fa una selecció d'alarmes de la taula Alarmes segons una condicio
208  * @param condicio Condicio de la selecció
209  * @return Llista d'alarmes seleccionades
210  */
211 public List<Alarma> SelectAlarmes(String condicio){
212     List<Alarma> Alarmes = new ArrayList<Alarma>();
213     try {
214         ResultSet RS;
215         Statement S = C.createStatement();
216         if (condicio==null){
217             RS = S.executeQuery("SELECT * FROM Alarmes");
218         } else {
219             RS = S.executeQuery("SELECT * FROM Alarmes WHERE " + condicio);
220         }
221         while (RS.next())
222             Alarmes.add(new Alarma(RS.getString("CodiHistorial"),RS.getTimestamp("DataAlarma")));
223     } catch (SQLException ex) {}
224     return Alarmes;
225 }
226 /**
227  * Insereix una alarma a la taula Alarmes
228  * @param Alarma Alarma a inserir
229  */
230 public void InsertAlarma(Alarma Alarma){
231     try {
232         PreparedStatement PS = C.prepareStatement("INSERT INTO Alarmes (
233             CodiHistorial,DataAlarma) VALUES (?,?)");
234         PS.setString(1, Alarma.getCodiHistorial());
235         PS.setTimestamp(2, Utils.obtenirTimestampSQL(Alarma.getDataAlarma()));
236         PS.executeUpdate();
237     } catch (SQLException ex) {}
238 }
239 /**
240  * Elimina totes les alarmes que compleixen una condició
241  * @param condicio Condicio d'eliminació
242  */
243 public void DeleteAlarmes(String condicio){

```

```

245     try {
246         Statement S = C.createStatement();
247         S.executeUpdate("DELETE FROM Alarmes WHERE " + condicio);
248     } catch (SQLException ex) {
249         Logger.getLogger(DB.class.getName()).log(Level.SEVERE, null, ex);
250     }
251 }
252
253 /**
254  * Genera el recompte d'alarmes mensual segons una condicio
255  * @param condicio Condicio de la selecció
256  * @return Parella Mes - Num d'alarmes
257  */
258 public List<Map<String,Integer>> SelectAlarmesPerMes(String condicio){
259     List<Map<String,Integer>> LR = new ArrayList<Map<String,Integer>>();
260     Map<String,Integer> HM = null;
261     try {
262         ResultSet RS;
263         Statement S = C.createStatement();
264         RS = S.executeQuery("SELECT MONTH(DataAlarma), YEAR(DataAlarma), COUNT(*)
265                               FROM Alarmes WHERE " + condicio + " GROUP BY YEAR(DataAlarma), MONTH(
266                               DataAlarma)");
267         while (RS.next()){
268             HM = new HashMap<String,Integer>();
269             String Mes = RS.getString(1);
270             Mes = ((Integer.valueOf(Mes)<10) ? "0":"" ) + Mes;
271             HM.put(Mes+"/"+RS.getString(2),RS.getInt(3));
272             LR.add(HM);
273         }
274     } catch (SQLException ex) {}
275     return LR;
276 }
277
278 /**
279  * Genera el recompte d'alarmes donada una condicio
280  * @param condicio Condicio de la selecció
281  * @return Nombre d'alarmes
282  */
283 public Integer SelectNumAlarmes(String condicio){
284     int numAlarmes = 0;
285     try {
286         ResultSet RS;
287         Statement S = C.createStatement();
288         RS = S.executeQuery("SELECT COUNT(*) FROM Alarmes WHERE " + condicio);
289         RS.next();
290         numAlarmes = RS.getInt(1);
291     } catch (SQLException ex) {}
292     return numAlarmes;
293 }

```

## Codi D.7: (AppSeguiment) Historial.java

```

1 package Dades;
2
3 import java.io.ByteArrayInputStream;
4 import java.io.ByteArrayOutputStream;
5 import java.io.DataInputStream;
6 import java.io.DataOutputStream;
7 import java.io.IOException;
8 import java.util.Enumeration;
9 import java.util.Vector;
10
11 /**
12  * Classe representativa de l'Historial d'Alarmes
13  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
14  * @version 1.0 (Juny de 2010)
15  */

```

```

16 public class Historial {
17
18     /** Vector de timestamps */
19     private Vector V;
20     /** Identificador de la TimeZone */
21     private String TZid;
22
23     /**
24      * Crea una instància de Historial
25      */
26     public Historial(){
27         V = new Vector();
28     }
29
30     /**
31      * Obté el vector amb l'Historial d'Alarmes
32      * @return Vector de timestamps
33      */
34     public Vector getHistorial(){
35         return V;
36     }
37
38     /**
39      * Guarda el vector amb l'Historial d'Alarmes
40      * @param valor      Vector de timestamps
41      */
42     public void setHistorial(Vector valor){
43         V = valor;
44     }
45
46     /**
47      * Reseteja l'Historial d'Alarmes
48      */
49     public void delHistorial(){
50         V.removeAllElements();
51     }
52
53     /**
54      * Obtenir l'identificador de la TimeZone
55      */
56     public String getTZid(){
57         return TZid;
58     }
59
60     /**
61      * Serialitza l'objecte
62      * @return Stream de bytes (Objecte serialitzat)
63      */
64     public byte[] toByteStream() throws IOException{
65         ByteArrayOutputStream baos = new ByteArrayOutputStream();
66         DataOutputStream dos = new DataOutputStream(baos);
67         dos.writeUTF(TZid);
68         dos.writeInt(V.size());
69         Enumeration vEnum = V.elements();
70         while(vEnum.hasMoreElements())
71             dos.writeUTF((String)vEnum.nextElement());
72         dos.flush();
73         return baos.toByteArray();
74     }
75
76     /**
77      * Des-Serialitza l'objecte
78      * @param dades      Stream de bytes (Objecte des-serialitzat)
79      */
80     public void fromByteStream(byte[] dades) throws IOException{
81         ByteArrayInputStream bais = new ByteArrayInputStream(dades);
82         DataInputStream dis = new DataInputStream(bais);
83         int n = dis.readInt();
84         TZid = dis.readUTF();
85         V = new Vector();

```

```

86         while ((n--)>0)
87             V.addElement(dis.readUTF());
88     }
89 }
90 }

```

## Codi D.8: (AppSeguiment) Model.java

```

1 package Dades;
2
3 import Constants.CONST;
4 import Constants.LANG;
5 import Utils.Utils;
6 import java.text.SimpleDateFormat;
7 import java.util.ArrayList;
8 import java.util.Calendar;
9 import java.util.Enumeration;
10 import java.util.HashMap;
11 import java.util.List;
12 import java.util.Map;
13 import java.util.Observable;
14 import java.util.TimeZone;
15 import java.util.Vector;
16
17 /**
18  * Classe que conté totes les dades de l'aplicació. Actua com a Model dins del patró MVC
19  * (Model-View-Controller)
20  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
21  * @version 1.0 (Juny de 2010)
22  */
23 public class Model extends Observable {
24     /** Objecte Gestor de la BD */
25     private DB DB;
26     /** Idioma de la interfície de l'aplicació */
27     private int Idioma;
28     /** Llista de pacients donats d'alta */
29     private List<Pacient> PacientsActuals;
30     /** Pacient que s'està analitzant */
31     private Pacient PacientActual;
32
33     /**
34      * Crea una instància de Model
35      */
36     public Model(){
37         DB = new DB();
38         DB.Connectar();
39         Idioma = DB.SelectIdioma();
40         PacientsActuals = DB.SelectPacients(null);
41         DB.Desconnectar();
42     }
43
44     /**
45      * Obté l'idioma de la interfície de l'aplicació
46      * @return Identificador de l'idioma
47      */
48     public int getIdioma(){
49         return Idioma;
50     }
51
52     /**
53      * Estableix l'idioma de la interfície de l'aplicació
54      * @param valor Identificador de l'idioma
55      */
56     public void setIdioma(int Idioma){
57         this.Idioma = Idioma;
58         DB.Connectar();
59         DB.UpdateIdioma(Idioma);
60         DB.Desconnectar();

```

```

61         setChanged();
62         notifyObservers();
63     }
64
65     /**
66     * Obte la llista de tots els pacients
67     * @return Llista de pacients
68     */
69     public List<Pacient> getPacients(){
70         return PacientsActuals;
71     }
72
73     /**
74     * Obte el pacient amb un cert codi d'historial
75     * @param CodiHistorial Codi d'historial del pacient
76     * @return Pacient seleccionat
77     */
78     public Pacient getPacient(String CodiHistorial){
79         DB.Connectar();
80         List<Pacient> Pacients = DB.SelectPacients("CodiHistorial='"+CodiHistorial+"'");
81         DB.Desconnectar();
82         return ((Pacients.size()>0) ? Pacients.get(0) : null);
83     }
84
85     /**
86     * Estableix un nou pacient a l'aplicació
87     * @param Pacient Pacient a inserir
88     */
89     public void setPacient(Pacient Pacient){
90         DB.Connectar();
91         DB.InsertPacient(Pacient);
92         PacientsActuals = DB.SelectPacients(null);
93         DB.Desconnectar();
94         setChanged();
95         notifyObservers();
96     }
97
98     /**
99     * Actualitza un pacient existent en l'aplicació
100    * @param Pacient Pacient a actualitzar
101    * @param CodiHistorial Codi d'historial del pacient a modificar
102    */
103    public void updPacient(Pacient Pacient, String CodiHistorial){
104        DB.Connectar();
105        DB.UpdatePacient(Pacient,"CodiHistorial='"+CodiHistorial+"'");
106        PacientsActuals = DB.SelectPacients(null);
107        DB.Desconnectar();
108        setChanged();
109        notifyObservers();
110    }
111
112    /**
113    * Elimina un pacient (i les seves alarmes) existent en l'aplicació
114    * @param Pacient Pacient a eliminar
115    */
116    public void delPacient(Pacient Pacient){
117        DB.Connectar();
118        DB.DeletePacient("CodiHistorial='"+Pacient.getCodiHistorial()+"'");
119        PacientsActuals = DB.SelectPacients(null);
120        DB.Desconnectar();
121        setChanged();
122        notifyObservers();
123    }
124
125    /**
126    * Obte les alarmes d'un pacient en concret, donada una condició
127    * @param Pacient Pacient en qüestió
128    * @param condicio Condició de les alarmes a obtenir
129    */
130    public List<Alarma> getAlarmes(Pacient Pacient, String condicio){

```

```

131     DB.Connectar();
132     if (Pacient == null)
133         return (new ArrayList<Alarma>());
134     condicio = (condicio==null) ? "CodiHistorial='"+Pacient.getCodiHistorial()+"' " :
135         "CodiHistorial='"+Pacient.getCodiHistorial()+"' AND "+condicio;
136     List<Alarma> Alarmes = DB.SelectAlarmes(condicio);
137     DB.Desconnectar();
138     return Alarmes;
139 }
140 /**
141  * Afegeix una nova alarma a l'aplicació
142  * @param Alarma Alarma a afegir
143  */
144 public void setAlarma(Alarma Alarma){
145     DB.Connectar();
146     DB.InsertAlarma(Alarma);
147     DB.Desconnectar();
148     setChanged();
149     notifyObservers();
150 }
151
152 public List<Map<String,Integer>> getDadesGBarresMensual(Pacient Pacient){
153     DB.Connectar();
154     List<Map<String,Integer>> LR = DB.SelectAlarmesPerMes("CodiHistorial='"+Pacient.
155         getCodiHistorial()+"'");
156     DB.Desconnectar();
157     return LR;
158 }
159
160 public List<Map<String,Integer>> getDadesGBarresUltimMes(Pacient Pacient){
161     List<Map<String,Integer>> LR = null;
162     List<Alarma> LA = getAlarmes(Pacient,null);
163     if (!LA.isEmpty()){
164         Alarma Alarma = LA.get(LA.size()-1);
165         Calendar cal = Calendar.getInstance(TimeZone.getDefault());
166         cal.setTime(Alarma.getDataAlarma());
167         LR = new ArrayList<Map<String,Integer>>();
168         int numAlarmes;
169         DB.Connectar();
170         cal.add(Calendar.DATE,-30);
171         for (int i=0; i<30; i++){
172             cal.add(Calendar.DATE, +1);
173             numAlarmes = DB.SelectNumAlarmes("CodiHistorial='"+Pacient.
174                 getCodiHistorial()+"' AND DAY(DataAlarma)="+cal.get(Calendar.
175                 DAY_OF_MONTH)+" AND MONTH(DataAlarma)="+cal.get(Calendar.MONTH)+1)+
176                 " AND YEAR(DataAlarma)="+cal.get(Calendar.YEAR));
177             Map<String,Integer> HM = new HashMap<String,Integer>();
178             HM.put(Utils.obtenirDataTXT(cal.getTime(), CONST.DATA_CURTA, Idioma),
179                 numAlarmes);
180             LR.add(HM);
181         }
182         DB.Desconnectar();
183     }
184     return LR;
185 }
186
187 public List<Map<String,Double>> getDadesGBarresPerHores(Pacient Pacient){
188     List<Map<String,Double>> LR = new ArrayList<Map<String,Double>>();
189     int numTotalAlarmes, numAlarmes, tmp;
190     double percent;
191     String Hora;
192     DB.Connectar();
193     numTotalAlarmes = DB.SelectNumAlarmes("CodiHistorial='"+Pacient.getCodiHistorial
194         ()+"'");
195     for (int i=0; i<24; i++){
196         numAlarmes = DB.SelectNumAlarmes("CodiHistorial='"+Pacient.getCodiHistorial
197             ()+"' AND HOUR(DataAlarma)="+i);
198         Map<String,Double> HM = new HashMap<String,Double>();
199         Hora = ((i<10) ? "0":"") + String.valueOf(i);

```

```

193         tmp = ((int)(((double)numAlarmes/numTotalAlarmes)*100*10);
194         percent = ((double)tmp/10);
195         HM.put(Hora+"h",percent);
196         LR.add(HM);
197     }
198     DB.Desconnectar();
199     return LR;
200 }
201
202 @SuppressWarnings("empty-statement")
203 public List<Map<String,Integer>> getDadesGFormatgeDiurnNocturn(Pacient Pacient){
204     List<Map<String,Integer>> LR = new ArrayList<Map<String,Integer>>();
205     int numAlarmes;
206     int numAlarmesDiurnes=0, numAlarmesNocturnes=0;
207     String Hora;
208     DB.Connectar();
209     for (int i=0; i<24; i++){
210         numAlarmes = DB.SelectNumAlarmes("CodiHistorial='"+Pacient.getCodiHistorial
211             ()+"' AND HOUR(DataAlarma)="+i);
212         if (i>8 && i<22){
213             numAlarmesDiurnes += numAlarmes;
214         } else {
215             numAlarmesNocturnes += numAlarmes;
216         }
217     }
218     Map<String,Integer> HM = new HashMap<String,Integer>();
219     HM.put(LANG.TXT_DIURNES[Idioma],numAlarmesDiurnes*100/(numAlarmesDiurnes +
220         numAlarmesNocturnes));
221     LR.add(HM);
222     HM = new HashMap<String,Integer>();
223     HM.put(LANG.TXT_NOCTURNES[Idioma],numAlarmesNocturnes*100/(numAlarmesDiurnes +
224         numAlarmesNocturnes));
225     LR.add(HM);
226     DB.Desconnectar();
227     return LR;
228 }
229
230 /**
231  * Estableix el pacient que s'esta analitzant en aquest moment
232  * @param PacientActual Pacient que s'està analitzant
233  */
234 public void setPacientActual(Pacient PacientActual){
235     this.PacientActual = PacientActual;
236     setChanged();
237     notifyObservers();
238 }
239
240 /**
241  * Obté el pacient que s'està analitzant en aquest moment
242  * @return Pacient que s'esta analitzant
243  */
244 public Pacient getPacientActual(){
245     return PacientActual;
246 }
247
248 /**
249  * Insereix al sistema les alarmes contingudes en un objecte Historial
250  * @param Historial Historial d'alarmes
251  * @return Nombre d'alarmes rebudes en l'historial
252  */
253 public int setHistorial(Pacient P, Historial H){
254     Vector v = H.getHistorial();
255     Enumeration vEnum = v.elements();
256     while (vEnum.hasMoreElements())
257         setAlarma(new Alarma(P.getCodiHistorial(),Utils.obtenirData(Long.parseLong(
258             vEnum.nextElement().toString()),H.getTZid())));
259     if (v.size()>0){
260         P.setDarreraTransferencia(Utils.obtenirDataActual());
261         updPacient(P,P.getCodiHistorial());
262     }
263 }

```

```

259         return v.size();
260     }
261 }
262 }

```

### Codi D.9: (AppSeguiment) Pacient.java

```

1
2 package Dades;
3
4 import java.util.Date;
5
6 /**
7  * Classe representativa d'un pacient
8  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
9  * @version 1.0 (Juny de 2010)
10 */
11 public class Pacient {
12
13     /** Codi de l'historial PPStop del pacient */
14     private String CodiHistorial;
15     /** Nom del pacient */
16     private String Nom;
17     /** Cognoms del pacient */
18     private String Cognoms;
19     /** Data de naixement del pacient */
20     private Date DataNaixement;
21     /** Sexe del pacient */
22     private int Sexe;
23     /** Data d'alta del pacient */
24     private Date DataAlta;
25     /** Data de la darrera transferència d'Historial */
26     private Date DarreraTransferencia;
27
28     /**
29      * Crea una instància de Pacient
30      * @param CodiHistorial Codi de l'historial PPStop del pacient
31      * @param Nom Nom del pacient
32      * @param Cognoms Cognoms del pacient
33      * @param Edat Edat del pacient
34      * @param Sexe Sexe del pacient
35      * @param DataAlta Data d'alta del pacient
36      * @param DarreraTransferencia Data de la darrera transferència d'Historial
37      */
38     public Pacient(String CodiHistorial, String Nom, String Cognoms, Date DataNaixement,
39                     int Sexe, Date DataAlta, Date DarreraTransferencia){
40         this.CodiHistorial = CodiHistorial;
41         this.Nom = Nom;
42         this.Cognoms = Cognoms;
43         this.DataNaixement = DataNaixement;
44         this.Sexe = Sexe;
45         this.DataAlta = DataAlta;
46         this.DarreraTransferencia = DarreraTransferencia;
47     }
48
49     /**
50      * Estableix les dades del pacient
51      * @param CodiHistorial Codi de l'historial PPStop del pacient
52      * @param Nom Nom del pacient
53      * @param Cognoms Cognoms del pacient
54      * @param Edat Edat del pacient
55      * @param Sexe Sexe del pacient
56      */
57     public void ActualitzaDades(String CodiHistorial, String Nom, String Cognoms, Date
58                                 DataNaixement, int Sexe){
59         this.CodiHistorial = CodiHistorial;
60         this.Nom = Nom;
61         this.Cognoms = Cognoms;
62         this.DataNaixement = DataNaixement;

```



```

61         this.Sexe = Sexe;
62     }
63
64     /**
65      * Estableix el codi de l'historial PPStop del pacient
66      * @param CodiHistorial    Codi de l'historial PPStop del pacient
67      */
68     public void setCodiHistorial(String CodiHistorial){
69         this.CodiHistorial = CodiHistorial;
70     }
71
72     /**
73      * Obté el codi de l'historial PPStop del pacient
74      * @return Codi de l'historial PPStop del pacient
75      */
76     public String getCodiHistorial(){
77         return CodiHistorial;
78     }
79
80     /**
81      * Estableix el nom del pacient
82      * @param Nom    Nom del pacient
83      */
84     public void setNom(String Nom){
85         this.Nom = Nom;
86     }
87
88     /**
89      * Obté el nom del pacient
90      * @return Nom del pacient
91      */
92     public String getNom(){
93         return Nom;
94     }
95
96     /**
97      * Estableix els cognoms del pacient
98      * @param Cognoms    Cognoms del pacient
99      */
100     public void setCognoms(String Cognoms){
101         this.Cognoms = Cognoms;
102     }
103
104     /**
105      * Obté els cognoms del pacient
106      * @return Cognoms del pacient
107      */
108     public String getCognoms(){
109         return Cognoms;
110     }
111
112     /**
113      * Estableix la data de naixement del pacient
114      * @param DataNaixement Data de naixement del pacient
115      */
116     public void setDataNaixement(Date DataNaixement){
117         this.DataNaixement = DataNaixement;
118     }
119
120     /**
121      * Obté la data de naixement del pacient
122      * @return Data de naixement del pacient
123      */
124     public Date getDataNaixement(){
125         return DataNaixement;
126     }
127
128     /**
129      * Estableix el sexe del pacient
130      * @param Sexe    Sexe del pacient

```

```

131     */
132     public void setSexe(int Sexe){
133         this.Sexe = Sexe;
134     }
135
136     /**
137     * Obté el sexe del pacient
138     * @return Sexe del pacient
139     */
140     public int getSexe(){
141         return Sexe;
142     }
143
144     /**
145     * Estableix la data d'alta del pacient
146     * @param DataAlta Data d'alta del pacient
147     */
148     public void setDataAlta(Date DataAlta){
149         this.DataAlta = DataAlta;
150     }
151
152     /**
153     * Obté la data d'alta del pacient
154     * @return Data d'alta del pacient
155     */
156     public Date getDataAlta(){
157         return DataAlta;
158     }
159
160     /**
161     * Estableix la data de la darrera transferència d'historial
162     * @param DarreraTransferencia Data de la darrera transferència d'Historial
163     */
164     public void setDarreraTransferencia(Date DarreraTransferencia){
165         this.DarreraTransferencia = DarreraTransferencia;
166     }
167
168     /**
169     * Obté la data de la darrera transferència d'historial
170     * @return Data de la darrera transferència d'historial
171     */
172     public Date getDarreraTransferencia(){
173         return DarreraTransferencia;
174     }
175
176     /**
177     * Representació textual de la classe Pacient, per a funcions de debugging
178     * @return Representacio textual de la classe
179     */
180     @Override
181     public String toString(){
182         return ("CodiHistorial:"+CodiHistorial+" / Nom:"+Nom+" / Cognoms:"+Cognoms+" /
183             DataNaixement:"+DataNaixement+" / Sexe:"+Sexe+" / DataAlta:"+DataAlta+" /
184             DarreraTransferencia:"+DarreraTransferencia);
185     }

```

## Codi D.10: (AppSeguiment) App.java

```

1 package Main;
2
3 import Bluetooth.Bluetooth;
4 import Dades.Model;
5 import Dades.Pacient;
6 import Utils.Informe;
7 import Vista.GUI;
8 import Vista.SplashScreen;
9

```

```

10 /**
11  * Classe principal de l'aplicació de seguiment PPStop
12  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
13  * @version 1.0 (Juny de 2010)
14  */
15 public class App {
16
17     /** Model de dades */
18     private static Model M;
19     /** Graphic User Interface (GUI) de l'aplicació */
20     private static GUI V;
21     /** Objecte d'obtenció d'historial a través de Bluetooth */
22     private static Bluetooth BT;
23     /** Objecte de generació d'informe PDF */
24     private static Informe I;
25
26     /**
27      * Obté l'historial d'alarmes del pacient especificat
28      * @param P      Pacient del que es vol obtenir l'historial
29      */
30     public void obtenirHistorial(Pacient P){
31         BT = new Bluetooth(V, M, P);
32     }
33
34     /**
35      * Genera l'informe PDF del pacient especificat
36      * @param P      Pacient del que es vol obtenir l'historial
37      * @param path   Path on generar l'informe PDF
38      */
39     public void generarInforme(Pacient P, String path) throws Exception{
40         I = new Informe(M, P, path);
41     }
42
43     /**
44      * Codi principal de l'aplicació
45      * @param args   Paràmetres de l'aplicació
46      */
47     public static void main(String[] args) {
48
49         final App A = new App();
50         M = new Model();
51         try {
52             javax.swing.SwingUtilities.invokeLater(new Runnable() {
53                 public void run() {
54                     V = new GUI(A,M);
55                     SplashScreen SS = new SplashScreen(V);
56                     Thread T = new Thread(SS);
57                     T.start();
58                 }
59             });
60         } catch (Exception e) {
61             System.out.println("Error: " + e);
62         }
63     }
64 }
65
66 }

```

### Codi D.11: (AppSeguiment) Grafics.java

```

1
2 package Utils;
3
4 import java.awt.Color;
5 import java.awt.image.BufferedImage;
6 import java.util.Iterator;
7 import java.util.List;
8 import java.util.Map;
9 import org.jfree.chart.ChartFactory;

```

```

10 import org.jfree.chart.JFreeChart;
11 import org.jfree.chart.axis.CategoryAxis;
12 import org.jfree.chart.axis.CategoryLabelPositions;
13 import org.jfree.chart.axis.NumberAxis;
14 import org.jfree.chart.axis.ValueAxis;
15 import org.jfree.chart.labels.ItemLabelAnchor;
16 import org.jfree.chart.labels.ItemLabelPosition;
17 import org.jfree.chart.labels.StandardCategoryItemLabelGenerator;
18 import org.jfree.chart.plot.CategoryPlot;
19 import org.jfree.chart.plot.PiePlot;
20 import org.jfree.chart.plot.PlotOrientation;
21 import org.jfree.chart.renderer.category.BarRenderer;
22 import org.jfree.chart.renderer.category.StandardBarPainter;
23 import org.jfree.data.category.DefaultCategoryDataset;
24 import org.jfree.data.general.DefaultPieDataset;
25 import org.jfree.ui.TextAnchor;
26
27 /**
28  * Classe per a generar les gràfiques estadístiques dels pacients
29  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
30  * @version 1.0 (Juny de 2010)
31  */
32 public class Grafics {
33
34     /**
35      * Crea una instància de Grafics
36      */
37     public Grafics(){}
38
39     /**
40      * Genera una grafica de barres amb les dades proporcionades (enters)
41      * @param LR      Llista de parells a representar
42      * @return Imatge de la gràfica
43      */
44     public BufferedImage getGBarresEnters(List<Map<String,Integer>> LR){
45         DefaultCategoryDataset DS = new DefaultCategoryDataset();
46         Iterator it = LR.iterator();
47         Map<String,Integer> HM;
48         while (it.hasNext()){
49             HM = (Map<String,Integer>) it.next();
50             Iterator it2 = HM.keySet().iterator();
51             String key = (String) it2.next();
52             DS.setValue(HM.get(key), "Alarmes", key);
53         }
54         int i=0;
55         JFreeChart chart = ChartFactory.createBarChart("", "", "", DS, PlotOrientation.
56             VERTICAL, false, true, false);
57         CategoryPlot CP = chart.getCategoryPlot();
58         CP.setBackgroundPaint(new Color(240,245,255));
59         CP.setRangeGridlinePaint(Color.darkGray);
60         CP.setRangeGridlinesVisible(true);
61         CategoryAxis CA = CP.getDomainAxis();
62         CA.setAxisLineVisible(false);
63         ValueAxis VA = CP.getRangeAxis();
64         VA.setAxisLineVisible(false);
65         VA.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
66         BarRenderer BR = (BarRenderer) CP.getRenderer();
67         BR.setBarPainter(new StandardBarPainter());
68         BR.setDrawBarOutline(false);
69         BR.setShadowVisible(false);
70         BR.setSeriesPaint(0, new Color(133,176,250));
71         BR.setBaseItemLabelsVisible(true);
72         BR.setBaseItemLabelGenerator(new StandardCategoryItemLabelGenerator());
73         BR.setBasePositiveItemLabelPosition(new ItemLabelPosition(ItemLabelAnchor.
74             OUTSIDE12, TextAnchor.BASELINE_CENTER));
75         CA.setCategoryLabelPositions(CategoryLabelPositions.UP_90);
76         BufferedImage BI = chart.createBufferedImage(800, 400);
77         return BI;
78     }
79 }

```

```

78  /**
79  * Genera una grafica de barres amb les dades proporcionades (decimals)
80  * @param LR      Llista de parells a representar
81  * @return Imatge de la gràfica
82  */
83  public BufferedImage getGBarresDecimals(List<Map<String, Double>> LR) {
84      DefaultCategoryDataset DS = new DefaultCategoryDataset();
85      Iterator it = LR.iterator();
86      Map<String,Double> HM;
87      while (it.hasNext()){
88          HM = (Map<String,Double>) it.next();
89          Iterator it2 = HM.keySet().iterator();
90          String key = (String) it2.next();
91          DS.setValue(HM.get(key), "Alarmes", key);
92      }
93      int i=0;
94      JFreeChart chart = ChartFactory.createBarChart("", "", "", DS, PlotOrientation.
          VERTICAL, false, true, false);
95      CategoryPlot CP = chart.getCategoryPlot();
96      CP.setBackgroundPaint(new Color(240,245,255));
97      CP.setRangeGridlinePaint(Color.darkGray);
98      CP.setRangeGridlinesVisible(true);
99      CategoryAxis CA = CP.getDomainAxis();
100     CA.setAxisLineVisible(false);
101     ValueAxis VA = CP.getRangeAxis();
102     VA.setAxisLineVisible(false);
103     BarRenderer BR = (BarRenderer) CP.getRenderer();
104     BR.setBarPainter(new StandardBarPainter());
105     BR.setDrawBarOutline(false);
106     BR.setShadowVisible(false);
107     BR.setSeriesPaint(0, new Color(133,176,250));
108     BR.setBaseItemLabelsVisible(true);
109     BR.setBaseItemLabelGenerator(new StandardCategoryItemLabelGenerator());
110     BR.setBasePositiveItemLabelPosition(new ItemLabelPosition(ItemLabelAnchor.
        OUTSIDE12, TextAnchor.BASELINE_CENTER));
111     CA.setCategoryLabelPositions(CategoryLabelPositions.UP_90);
112     BufferedImage BI = chart.createBufferedImage(800, 400);
113     return BI;
114 }
115
116 /**
117 * Genera una grafica circular amb les dades proporcionades
118 * @param LR      Llista de parells a representar
119 * @return Imatge de la gràfica
120 */
121 public BufferedImage getGFormatge(List<Map<String,Integer>> LR){
122     DefaultPieDataset DS = new DefaultPieDataset();
123     Iterator it = LR.iterator();
124     Map<String,Integer> HM;
125     int i=0;
126     while (it.hasNext()){
127         HM = (Map<String,Integer>) it.next();
128         Iterator it2 = HM.keySet().iterator();
129         String key = (String) it2.next();
130         DS.setValue(key + " (" + HM.get(key) + "%)", HM.get(key));
131     }
132     JFreeChart chart = ChartFactory.createPieChart("", DS, false, true, false);
133     PiePlot P = (PiePlot) chart.getPlot();
134     P.setOutlinePaint(Color.GRAY);
135     P.setBackgroundPaint(new Color(240,245,255));
136     P.setShadowPaint(new Color(240,245,255));
137     P.setLabelBackgroundPaint(Color.WHITE);
138     P.setLabelShadowPaint(new Color(240,245,255));
139     P.setSectionPaint(0, new Color(133,176,250));
140     BufferedImage BI = chart.createBufferedImage(580, 267);
141     return BI;
142 }
143
144 }

```

## Codi D.12: (AppSeguiment) Informe.java

```

1 package Utils;
2
3
4 import Constants.CONST;
5 import Constants.LANG;
6 import Dades.Alarma;
7 import Dades.Model;
8 import Dades.Pacient;
9 import com.itextpdf.text.BaseColor;
10 import com.itextpdf.text.Document;
11 import com.itextpdf.text.Element;
12 import com.itextpdf.text.ExceptionConverter;
13 import com.itextpdf.text.Font;
14 import com.itextpdf.text.Font.FontFamily;
15 import com.itextpdf.text.Image;
16 import com.itextpdf.text.PageSize;
17 import com.itextpdf.text.Paragraph;
18 import com.itextpdf.text.Rectangle;
19 import com.itextpdf.text.pdf.BaseFont;
20 import com.itextpdf.text.pdf.ColumnText;
21 import com.itextpdf.text.pdf.PdfContentByte;
22 import com.itextpdf.text.pdf.PdfPCell;
23 import com.itextpdf.text.pdf.PdfPTable;
24 import com.itextpdf.text.pdf.PdfPageEventHelper;
25 import com.itextpdf.text.pdf.PdfTemplate;
26 import com.itextpdf.text.pdf.PdfWriter;
27 import java.awt.Color;
28 import java.io.FileOutputStream;
29 import java.util.Date;
30 import java.util.Iterator;
31 import java.util.List;
32
33 /**
34  * Classe per a generar l'informe de seguiment PPStop en PDF
35  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
36  * @version 1.0 (Juny de 2010)
37  */
38 public class Informe extends PdfPageEventHelper {
39
40     /** Referència al Model */
41     private Model M;
42     /** Pacient sobre el que es genera l'informe */
43     private Pacient Pacient;
44     /** Path on guardar l'informe PDF generat */
45     private String PathGuardar;
46     /** Template del PDF per a inserir el número de pagina */
47     private PdfTemplate NumPag;
48     /** Font base per al numero de pàgina */
49     private BaseFont FontNumPag;
50
51     /**
52      * Crea una instància d'Informe i genera l'informe PDF
53      */
54     public Informe(Model M, Pacient Pacient, String PathGuardar) throws Exception {
55         this.M = M;
56         this.Pacient = Pacient;
57         this.PathGuardar = PathGuardar;
58         generarInforme();
59     }
60
61     /**
62      * Genera l'informe PDF
63      */
64     public void generarInforme() throws Exception {
65         Document D = new Document(PageSize.A4);
66         PdfWriter W = PdfWriter.getInstance(D, new FileOutputStream(PathGuardar+".pdf"));
67         W.setViewerPreferences(PdfWriter.PageLayoutTwoColumnLeft);
68         W.setPageEvent(this);

```

```

69         crearMetadata(D);
70         D.open();
71         crearHeader(D);
72         crearTaulaDadesPacient(D);
73         crearTaulaGrafics(D);
74         crearTaulaAlarmes(D,W);
75         D.close();
76     }
77
78     /**
79     * Escriu els metadata del document
80     * @param Document Document de PDF
81     */
82     private void crearMetadata(Document D){
83         D.addTitle("Informe PPStop");
84         D.addAuthor("PPStop");
85     }
86
87     /**
88     * Escriu la capçalera del document
89     * @param Document Document de PDF
90     */
91     private void crearHeader(Document D) throws Exception {
92         Image I = Image.getInstance ("res/logo.png");
93         Paragraph P = new Paragraph(LANG.TXT_INFORME[M.getIdioma()],new Font(FontFamily.
94             HELVETICA, 20, Font.NORMAL));
95         PdfPTable T = new PdfPTable(2);
96         T.setWidthPercentage(100f);
97         T.setSpacingAfter(5f);
98         PdfPCell C = new PdfPCell(I);
99         C.setHorizontalAlignment(Element.ALIGN_LEFT);
100        C.setBorder(0);
101        T.addCell(C);
102        C = new PdfPCell(P);
103        C.setHorizontalAlignment(Element.ALIGN_RIGHT);
104        C.setVerticalAlignment(Element.ALIGN_MIDDLE);
105        C.setBorder(0);
106        T.addCell(C);
107        D.add(T);
108        P = new Paragraph(LANG.TXT_DATAGENERACIO[M.getIdioma()]+": "+Utils.
109            obtenirDataTXT(Utils.obtenirDataActual(),CONST.DATA_SENCERA,M.getIdioma()),
110            new Font(FontFamily.HELVETICA, 9, Font.ITALIC, new BaseColor(Color.GRAY)));
111        P.setAlignment(Element.ALIGN_RIGHT);
112        P.setSpacingAfter(10f);
113        D.add(P);
114    }
115
116    /**
117    * Escriu la taula de dades del pacient al document
118    * @param Document Document de PDF
119    */
120    private void crearTaulaDadesPacient(Document D) throws Exception {
121        Paragraph P = new Paragraph(LANG.TXT_DADESPACIENT[M.getIdioma()].toUpperCase(),
122            new Font(FontFamily.HELVETICA, 9, Font.BOLD));
123        float[] widths = {1f,4f};
124        PdfPTable T = new PdfPTable(widths);
125        T.setWidthPercentage(100f);
126        T.setSpacingAfter(10f);
127        PdfPCell C = new PdfPCell(P);
128        C.setColspan(2);
129        C.setPadding(3f);
130        C.setHorizontalAlignment(Element.ALIGN_LEFT);
131        C.setVerticalAlignment(Element.ALIGN_MIDDLE);
132        C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
133        C.setBorder(0);
134        T.addCell(C);
135        C = new PdfPCell(new Paragraph(LANG.TXT_COGNOMS[M.getIdioma()]+", "+LANG.TXT_NOM
136            [M.getIdioma()],new Font(FontFamily.HELVETICA, 9, Font.BOLD)));
137        C.setBorder(0);
138        C.setPadding(3f);

```

```

134     C.setPaddingTop(4f);
135     T.addCell(C);
136     C = new PdfPCell(new Paragraph(Pacient.getCognoms().toUpperCase()+" "+Pacient.
        getNom().toUpperCase(),new Font(FontFamily.HELVETICA, 9, Font.NORMAL)));
137     C.setBorder(0);
138     C.setPadding(3f);
139     C.setPaddingTop(4f);
140     T.addCell(C);
141     C = new PdfPCell(new Paragraph(LANG.TXT_EDAT[M.getIdioma()]+ " / " +LANG.TXT_SEXE[
        M.getIdioma()],new Font(FontFamily.HELVETICA, 9, Font.BOLD)));
142     C.setBorder(0);
143     C.setPadding(3f);
144     C.setPaddingTop(4f);
145     T.addCell(C);
146     C = new PdfPCell(new Paragraph(String.valueOf(Utills.obtenirEdat(Pacient.
        getDataNaixement()))+" "+LANG.TXT_ANYIS[M.getIdioma()]+ " / " +LANG.TXT_SEXETXT
        [M.getIdioma()][Pacient.getSexe()],new Font(FontFamily.HELVETICA, 9, Font.
        NORMAL)));
147     C.setBorder(0);
148     C.setPadding(3f);
149     C.setPaddingTop(4f);
150     T.addCell(C);
151     C = new PdfPCell(new Paragraph(LANG.TXT_DATAALTA[M.getIdioma()],new Font(
        FontFamily.HELVETICA, 9, Font.BOLD)));
152     C.setBorder(0);
153     C.setPadding(3f);
154     C.setPaddingTop(4f);
155     T.addCell(C);
156     C = new PdfPCell(new Paragraph(Utills.obtenirDataTXT(Pacient.getDataAlta(),CONST.
        DATA_CURTA,M.getIdioma()),new Font(FontFamily.HELVETICA, 9, Font.NORMAL)));
157     C.setBorder(0);
158     C.setPadding(3f);
159     C.setPaddingTop(4f);
160     T.addCell(C);
161     C = new PdfPCell(new Paragraph(LANG.TXT_DARRERATRANSFERENCIA[M.getIdioma()],new
        Font(FontFamily.HELVETICA, 9, Font.BOLD)));
162     C.setBorder(0);
163     C.setPadding(3f);
164     C.setPaddingTop(4f);
165     T.addCell(C);
166     C = new PdfPCell(new Paragraph(Utills.obtenirDataTXT(Pacient.
        getDarreraTransferencia(),CONST.DATA_SENCERA,M.getIdioma()),new Font(
        FontFamily.HELVETICA, 9, Font.NORMAL)));
167     C.setBorder(0);
168     C.setPadding(3f);
169     C.setPaddingTop(4f);
170     T.addCell(C);
171     D.add(T);
172 }
173
174 /**
175  * Escriu la taula de gràfics estadístics del pacient al document
176  * @param Document Document de PDF
177  */
178 private void crearTaulaGrafics(Document D) throws Exception{
179     Grafics G = new Grafics();
180     Image I1 = Image.getInstance(G.getGBarresEnters(M.getDadesGBarresMensual(Pacient
        )),null);
181     I1.scaleToFit(260, 250);
182     Image I2 = Image.getInstance(G.getGBarresEnters(M.getDadesGBarresUltimMes(
        Pacient)),null);
183     I2.scaleToFit(260, 250);
184     Image I3 = Image.getInstance(G.getGBarresDecimals(M.getDadesGBarresPerHores(
        Pacient)),null);
185     I3.scaleToFit(260, 250);
186     Image I4 = Image.getInstance(G.getGFormatge(M.getDadesGFormatgeDiurnNocturn(
        Pacient)),null);
187     I4.scaleToFit(255, 250);
188     Paragraph P = new Paragraph(LANG.TXT_GRAFIQUESESTADISTIQUES[M.getIdioma()].
        toUpperCase(),new Font(FontFamily.HELVETICA, 9, Font.BOLD));

```



```

189     float[] widths = {1f,1f};
190     PdfPTable T = new PdfPTable(widths);
191     T.setWidthPercentage(100f);
192     T.setSpacingAfter(10f);
193     PdfPCell C = new PdfPCell(P);
194     C.setColspan(2);
195     C.setPadding(3f);
196     C.setHorizontalAlignment(Element.ALIGN_LEFT);
197     C.setVerticalAlignment(Element.ALIGN_MIDDLE);
198     C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
199     C.setBorder(0);
200     T.addCell(C);
201     C = new PdfPCell(I1);
202     C.setPaddingTop(6f);
203     C.setHorizontalAlignment(Element.ALIGN_LEFT);
204     C.setBorder(0);
205     T.addCell(C);
206     C = new PdfPCell(I2);
207     C.setPaddingTop(6f);
208     C.setHorizontalAlignment(Element.ALIGN_RIGHT);
209     C.setBorder(0);
210     T.addCell(C);
211     P = new Paragraph(LANG.TXT_GRAFIC1[M.getIdioma()],new Font(FontFamily.HELVETICA,
212         9, Font.ITALIC));
213     C = new PdfPCell(P);
214     C.setHorizontalAlignment(Element.ALIGN_CENTER);
215     C.setBorder(0);
216     T.addCell(C);
217     P = new Paragraph(LANG.TXT_GRAFIC2[M.getIdioma()],new Font(FontFamily.HELVETICA,
218         9, Font.ITALIC));
219     C = new PdfPCell(P);
220     C.setHorizontalAlignment(Element.ALIGN_CENTER);
221     C.setBorder(0);
222     T.addCell(C);
223     C = new PdfPCell(I3);
224     C.setPaddingTop(6f);
225     C.setHorizontalAlignment(Element.ALIGN_LEFT);
226     C.setBorder(0);
227     T.addCell(C);
228     C = new PdfPCell(I4);
229     C.setPaddingTop(8f);
230     C.setHorizontalAlignment(Element.ALIGN_RIGHT);
231     C.setBorder(0);
232     T.addCell(C);
233     P = new Paragraph(LANG.TXT_GRAFIC3[M.getIdioma()],new Font(FontFamily.HELVETICA,
234         9, Font.ITALIC));
235     C = new PdfPCell(P);
236     C.setHorizontalAlignment(Element.ALIGN_CENTER);
237     C.setBorder(0);
238     T.addCell(C);
239     P = new Paragraph(LANG.TXT_GRAFIC4[M.getIdioma()],new Font(FontFamily.HELVETICA,
240         9, Font.ITALIC));
241     C = new PdfPCell(P);
242     C.setHorizontalAlignment(Element.ALIGN_CENTER);
243     C.setBorder(0);
244     T.addCell(C);
245     D.add(T);
246 }
247
248 /**
249  * Escriu la taula d'alarmes del pacient al document
250  * @param Document Document de PDF
251  * @param PdfWriter Escriptor de PDF
252  */
253 private void crearTaulaAlarmes(Document D, PdfWriter W) throws Exception {
254     Paragraph P = new Paragraph(LANG.TXT_HISTORIALALARMES[M.getIdioma()].toUpperCase(),new Font(FontFamily.HELVETICA, 9, Font.BOLD));
255     PdfPTable T = new PdfPTable(1);
256     T.setWidthPercentage(100f);
257     T.setSpacingAfter(3f);

```

```

254 PdfPCell C = new PdfPCell(P);
255 C.setPadding(3f);
256 C.setHorizontalAlignment(Element.ALIGN_LEFT);
257 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
258 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
259 C.setBorder(0);
260 T.addCell(C);
261 D.add(T);
262 ColumnText CT = new ColumnText(W.getDirectContent());
263 float[] widths = {2f,1f,2f,4f,2f,3f,4f};
264 T = new PdfPTable(widths);
265 T.setWidthPercentage(100f);
266 C = new PdfPCell(new Paragraph("N",new Font(FontFamily.HELVETICA, 9, Font.BOLD))
    );
267 C.setBorder(0);
268 C.setPadding(3f);
269 C.setHorizontalAlignment(Element.ALIGN_LEFT);
270 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
271 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
272 T.addCell(C);
273 C = new PdfPCell(new Paragraph("T",new Font(FontFamily.HELVETICA, 9, Font.BOLD))
    );
274 C.setBorder(0);
275 C.setPadding(3f);
276 C.setHorizontalAlignment(Element.ALIGN_LEFT);
277 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
278 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
279 T.addCell(C);
280 C = new PdfPCell(new Paragraph(LANG.TXT_ANY[M.getIdioma()],new Font(FontFamily.
    HELVETICA, 9, Font.BOLD)));
281 C.setBorder(0);
282 C.setPadding(3f);
283 C.setHorizontalAlignment(Element.ALIGN_LEFT);
284 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
285 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
286 T.addCell(C);
287 C = new PdfPCell(new Paragraph(LANG.TXT_MES[M.getIdioma()],new Font(FontFamily.
    HELVETICA, 9, Font.BOLD)));
288 C.setBorder(0);
289 C.setPadding(3f);
290 C.setHorizontalAlignment(Element.ALIGN_LEFT);
291 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
292 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
293 T.addCell(C);
294 C = new PdfPCell(new Paragraph(LANG.TXT_DIA[M.getIdioma()],new Font(FontFamily.
    HELVETICA, 9, Font.BOLD)));
295 C.setBorder(0);
296 C.setPadding(3f);
297 C.setHorizontalAlignment(Element.ALIGN_LEFT);
298 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
299 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
300 T.addCell(C);
301 C = new PdfPCell(new Paragraph(LANG.TXT_HORA[M.getIdioma()],new Font(FontFamily.
    HELVETICA, 9, Font.BOLD)));
302 C.setBorder(0);
303 C.setPadding(3f);
304 C.setHorizontalAlignment(Element.ALIGN_LEFT);
305 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
306 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
307 T.addCell(C);
308 C = new PdfPCell(new Paragraph("Dif.",new Font(FontFamily.HELVETICA, 9, Font.
    BOLD)));
309 C.setBorder(0);
310 C.setPadding(3f);
311 C.setHorizontalAlignment(Element.ALIGN_LEFT);
312 C.setVerticalAlignment(Element.ALIGN_MIDDLE);
313 C.setBackgroundColor(new BaseColor(new Color(200,220,255)));
314 T.addCell(C);
315 List<Alarma> Alarmes = M.getAlarmes(Pacient,null);
316 Iterator it = Alarmes.iterator();

```

```

317     Alarma Alarma;
318     int i = 0;
319     Date dataAnterior = null;
320     while (it.hasNext()){
321         Alarma = ((Alarma)it.next());
322         C = new PdfPCell(new Paragraph("#"+(++i),new Font(FontFamily.HELVETICA, 9,
323             Font.BOLD)));
324         C.setBorder(0);
325         C.setPadding(3f);
326         if (i%2==0) C.setBackgroundColor(new BaseColor(new Color(240,245,255)));
327         T.addCell(C);
328         String tipus;
329         Color cTipus;
330         if (Alarma.getDataAlarma().getHours()>8 && Alarma.getDataAlarma().getHours()
331             <22){
332             tipus = "D";
333             cTipus = Color.BLUE;
334         } else {
335             tipus = "N";
336             cTipus = Color.RED;
337         }
338         C = new PdfPCell(new Paragraph(tipus,new Font(FontFamily.HELVETICA, 9, Font.
339             NORMAL, new BaseColor(cTipus))));
340         C.setBorder(0);
341         C.setPadding(3f);
342         if (i%2==0) C.setBackgroundColor(new BaseColor(new Color(240,245,255)));
343         T.addCell(C);
344         C = new PdfPCell(new Paragraph(Utils.obtenirDataTXT(Alarma.getDataAlarma(),
345             CONST.DATA_ANY,M.getIdioma()),new Font(FontFamily.HELVETICA, 9, Font.
346             NORMAL)));
347         C.setBorder(0);
348         C.setPadding(3f);
349         if (i%2==0) C.setBackgroundColor(new BaseColor(new Color(240,245,255)));
350         T.addCell(C);
351         C = new PdfPCell(new Paragraph(Utils.obtenirDataTXT(Alarma.getDataAlarma(),
352             CONST.DATA_DIA,M.getIdioma()),new Font(FontFamily.HELVETICA, 9, Font.
353             NORMAL)));
354         C.setBorder(0);
355         C.setPadding(3f);
356         if (i%2==0) C.setBackgroundColor(new BaseColor(new Color(240,245,255)));
357         T.addCell(C);
358         C = new PdfPCell(new Paragraph(Utils.obtenirDataTXT(Alarma.getDataAlarma(),
359             CONST.DATA_HORA,M.getIdioma()),new Font(FontFamily.HELVETICA, 9, Font.
360             NORMAL)));
361         C.setBorder(0);
362         C.setPadding(3f);
363         if (i%2==0) C.setBackgroundColor(new BaseColor(new Color(240,245,255)));
364         T.addCell(C);
365         dataAnterior = Alarma.getDataAlarma();
366     }
367     CT.addElement(T);
368     boolean primer = true;
369     int column = 0;
370     float height = 0;
371     float [][] x = {{D.left(), D.left()+240},{D.right()-240, D.right()}};
372     int status = ColumnText.START_COLUMN;

```

```

374         while (ColumnText.hasMoreText(status)){
375             if (primer){
376                 CT.setSimpleColumn(x[column][0], D.bottom()+20, x[column][1], D.top()-
                    height-492);
377             } else {
378                 CT.setSimpleColumn(x[column][0], D.bottom()+20, x[column][1], D.top()-
                    height-10);
379             }
380             status = CT.go();
381             if (++column >= x.length) {
382                 primer=false;
383                 column = 0;
384                 D.newPage();
385             }
386         }
387     }
388
389     /**
390     * Mètode que s'executa en començar a escriure el document
391     * @param PdfWriter      Escriptor de PDF
392     * @param Document       Document de PDF
393     */
394     @Override
395     public void onOpenDocument(PdfWriter W, Document D) {
396         NumPag = W.getDirectContent().createTemplate(100, 100);
397         NumPag.setBoundingBox(new Rectangle(0, 0, 100, 100));
398         try {
399             FontNumPag = BaseFont.createFont(BaseFont.HELVETICA, BaseFont.
                WINANSI, BaseFont.NOT_EMBEDDED);
400         } catch (Exception e) {
401             throw new ExceptionConverter(e);
402         }
403     }
404
405     /**
406     * Mètode que s'executa en acabar d'escriure una pàgina del document
407     * @param PdfWriter      Escriptor de PDF
408     * @param Document       Document de PDF
409     */
410     @Override
411     public void onEndPage(PdfWriter W, Document D) {
412         PdfContentByte CB = W.getDirectContent();
413         CB.saveState();
414         String text = Utils.substituirStr(LANG.TXT_PAGINADE[M.getIdioma()], "{X}");
415         String.valueOf(W.getPageNumber());
416         float textBase = D.bottom();
417         float textSize = FontNumPag.getWidthPoint(text, 9);
418         CB.beginText();
419         CB.setFontAndSize(FontNumPag, 9);
420         CB.setTextMatrix(D.left()+(D.right()-D.left())/2 - textSize/2, textBase);
421         CB.showText(text);
422         CB.endText();
423         CB.addTemplate(NumPag, D.left()+(D.right()-D.left())/2 + textSize/2, textBase);
424         CB.restoreState();
425     }
426
427     /**
428     * Mètode que s'executa en acabar d'escriure tot el document
429     * @param PdfWriter      Escriptor de PDF
430     * @param Document       Document de PDF
431     */
432     @Override
433     public void onCloseDocument(PdfWriter W, Document D) {
434         NumPag.beginText();
435         NumPag.setFontAndSize(FontNumPag, 9);
436         NumPag.setTextMatrix(0, 0);
437         NumPag.showText(String.valueOf(W.getPageNumber() - 1));
438         NumPag.endText();
439     }

```

440  
441 }

### Codi D.13: (AppSeguiment) Utils.java

```
1
2 package Utils;
3
4 import Constants.CONST;
5 import Constants.LANG;
6 import java.sql.Timestamp;
7 import java.util.Calendar;
8 import java.util.Date;
9 import java.util.TimeZone;
10 import java.util.Vector;
11
12 /**
13  * Classe amb funcions útils d'àmbit general
14  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
15  * @version 1.0 (Juny de 2010)
16  */
17 public class Utils {
18
19     /**
20      * Obte el timestamps actual
21      * @return Timestamp actual
22      */
23     public static long obtenirTimestampActual(){
24         return System.currentTimeMillis();
25     }
26
27     /**
28      * Obté un objecte Date amb la data actual
29      * @return Data actual
30      */
31     public static Date obtenirDataActual(){
32         Calendar cal = Calendar.getInstance(TimeZone.getDefault());
33         return new Date(cal.getTimeInMillis());
34     }
35
36     /**
37      * Obté un objecte Date a partir d'un timestamp i un identificador de TimeZone
38      * @param timestamp      Timestamp a convertir
39      * @param TZid            Identificador de la TimeZone
40      * @return Objecte Date
41      */
42     public static Date obtenirData(long timestamp, String TZid){
43         Calendar cal = Calendar.getInstance(TimeZone.getTimeZone(TZid));
44         cal.setTime(new Date(timestamp));
45         return cal.getTime();
46     }
47
48     /**
49      * Obté un objecte Date amb la data actual
50      * @return Data actual
51      */
52     public static Timestamp obtenirTimestampSQL(Date data){
53         return new Timestamp(data.getTime());
54     }
55
56     /**
57      * Obté la data en un format de text determinat a partir d'un Date
58      * @param timestamp      Timestamp a convertir
59      * @return Data formatada
60      */
61     public static String obtenirDataTXT(Date date, int tipus, int idioma){
62         Calendar cal = Calendar.getInstance(TimeZone.getDefault());
63         cal.setTime(date);
64         String Any = String.valueOf(cal.get(Calendar.YEAR));
```

```

65     String Mes = ((cal.get(Calendar.MONTH)+1)<10?"0":"") + String.valueOf(cal.get(
66         Calendar.MONTH)+1);
67     String Dia = (cal.get(Calendar.DAY_OF_MONTH)<10?"0":"") + String.valueOf(cal.get(
68         Calendar.DAY_OF_MONTH));
69     String Hora = (cal.get(Calendar.HOUR_OF_DAY)<10?"0":"") + String.valueOf(cal.get(
70         Calendar.HOUR_OF_DAY));
71     String Minut = (cal.get(Calendar.MINUTE)<10?"0":"") + String.valueOf(cal.get(
72         Calendar.MINUTE));
73     switch(tipus){
74         case CONST.DATA_ANY:
75             return Any;
76         case CONST.DATA_MES:
77             return Mes;
78         case CONST.DATA_MESTXT:
79             return LANG.TXT_MESTXT[idioma][Integer.parseInt(Mes)];
80         case CONST.DATA_DIA:
81             return Dia;
82         case CONST.DATA_HORA:
83             return (Hora + ":" + Minut + "h");
84         case CONST.DATA_SENCERA:
85             return (Dia + "/" + Mes + "/" + Any + " - " + Hora + ":" + Minut + "h");
86         case CONST.DATA_CURTA:
87             return (Dia + "/" + Mes + "/" + Any.substring(2,4));
88         case CONST.DATA_MENSUAL:
89             return (Mes + "/" + Any);
90         default:
91             return null;
92     }
93 }
94
95 /**
96  * Obté la diferència entre dues dates en format TXT
97  * @param timestamp      Timestamp de la data passada
98  * @param idioma          Idioma desitjat
99  * @return                Diferència en format TXT
100 */
101 public static String obtenirDiferenciaData(Date dataActual, Date dataAntiga, int
102     idioma){
103     if (dataAntiga == null) return "--";
104     long dif = dataActual.getTime() - dataAntiga.getTime();
105     if (dif < 60*1000){
106         return LANG.TXT_MENYSUNMINUT[idioma];
107     } else if (dif < 120*1000){
108         return (" +1 " + LANG.TXT_1MINUT[idioma]);
109     } else if (dif < 60*60*1000){
110         return (" + " + String.valueOf((int)Math.floor(dif/(60*1000))) + " " + LANG.
111             TXT_XMINUTS[idioma]);
112     } else if (dif < 120*60*1000){
113         return (" +1 " + LANG.TXT_1HORA[idioma]);
114     } else if (dif < 24*60*60*1000){
115         return (" + " + String.valueOf((int)Math.floor(dif/(3600*1000))) + " " + LANG.
116             TXT_XHORES[idioma]);
117     } else if (dif < 48*60*60*1000){
118         return (" +1 " + LANG.TXT_1DIA[idioma]);
119     } else {
120         return (" + " + String.valueOf((int)Math.floor(dif/(86400*1000))) + " " + LANG.
121             .TXT_XDIES[idioma]);
122     }
123 }
124
125 /**
126  * Obté l'edat donada la data de naixement
127  * @param dataNaixement  Data de naixement
128  * @return                Edat (anys)
129 */
130 public static int obtenirEdat(Date dataNaixement){
131     Calendar CdataActual = Calendar.getInstance();
132     Calendar CdataNaixement = Calendar.getInstance();
133     CdataNaixement.setTime(dataNaixement);
134     int dif_any = CdataActual.get(Calendar.YEAR) - CdataNaixement.get(Calendar.YEAR)

```

```

127         ;
128         int dif_mes = CdataActual.get(Calendar.MONTH) - CdataNaixement.get(Calendar.
MONTH);
129         int dif_dia = CdataActual.get(Calendar.DAY_OF_MONTH) - CdataNaixement.get(
Calendar.DAY_OF_MONTH);
130         if (dif_mes < 0 || (dif_mes == 0 && dif_dia < 0)) dif_any--;
131         return dif_any;
132     }
133
134     /**
135     * Inverteix els elements d'un vector
136     * @param vec    Vector a invertir
137     * @return      Vector invertit
138     */
139     public static Vector invertirVector(Vector vec){
140         Object temp;
141         for(int i=0; i < vec.size()/2; i++){
142             temp = vec.elementAt(vec.size()-(i+1));
143             vec.setElementAt(vec.elementAt(i), vec.size()-(i+1));
144             vec.setElementAt(temp, i);
145         }
146         return vec;
147     }
148
149     /**
150     * Reemplaça strings d'una cadena de caràcters
151     * @param text    Text sobre el qual s'aplica la substitució
152     * @param comodi  Cadena de text a substituir
153     * @param substitut Cadena de text substituïda
154     * @return      Text amb la substitució aplicada
155     */
156     public static String substituirStr(String text, String comodi, String substitut) {
157         StringBuffer SB = new StringBuffer();
158         int startPos = 0;
159         int searchStringPos = text.indexOf(comodi);
160         int searchStringLength = comodi.length();
161         while(searchStringPos!=-1){
162             SB.append(text.substring(startPos,searchStringPos)).append(substitut);
163             startPos = searchStringPos + searchStringLength;
164             searchStringPos = text.indexOf(comodi,startPos);
165         }
166         return SB.append(text.substring(startPos,text.length())).toString();
167     }
168 }

```

#### Codi D.14: (AppSeguiment) UtilsVista.java

```

1 package Utils;
2
3
4 import java.awt.Graphics2D;
5 import java.awt.RenderingHints;
6 import java.awt.image.BufferedImage;
7
8 /**
9  * Classe amb funcions útils per a la GUI
10  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
11  * @version 1.0 (Juny de 2010)
12  */
13 public class UtilsVista {
14
15     /**
16     * Retorna una imatge escalada a les dimensions proporcionades
17     * @param BI      Imatge a escalar
18     * @param width    Amplada desitxada (px.)
19     * @param height   Alçada desitxada (px.)
20     * @return Imatge escalada
21     */

```

```

22     public static BufferedImage EscalarImatge(BufferedImage BI, int width, int height){
23         BufferedImage I = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
24         Graphics2D graphics2D = I.createGraphics();
25         graphics2D.setRenderingHint(RenderingHints.KEY_INTERPOLATION, RenderingHints.
                VALUE_INTERPOLATION_BILINEAR);
26         graphics2D.drawImage(BI, 0, 0, width, height, null);
27         graphics2D.dispose();
28         return I;
29     }
30
31 }

```

### Codi D.15: (AppSeguiment) DialogPacient.java

```

1
2 package Vista;
3
4 import Constants.CONST;
5 import Constants.LANG;
6 import Dades.Model;
7 import Dades.Pacient;
8 import Utils.Utils;
9 import java.text.SimpleDateFormat;
10 import java.util.Date;
11 import java.util.Observable;
12 import javax.swing.ImageIcon;
13 import javax.swing.JDialog;
14 import javax.swing.JFrame;
15 import javax.swing.JOptionPane;
16
17 /**
18  * Classe que conté tota la GUI per a donar d'alta un nou pacient o modificar-ne un de
19  * ja existent
20  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
21  * @version 1.0 (Juny de 2010)
22  */
23 public class DialogPacient extends JDialog implements Vista {
24
25     /** Referència al model de dades */
26     private Model M;
27
28     /** Tipus d'acció a realitzar: crear nou pacient o modificar pacient que s'està tractant */
29     private int accio;
30
31     /**
32      * Crea una nova instància de DialogPacient
33      * @param GUI Referència a la GUI principal
34      * @param M Referència al Model de dades
35      * @param accio Indica si es tracta d'una alta de pacient o una modificació de dades de pacient
36      * @param titol Titol de la finestra
37      */
38     public DialogPacient(JFrame GUI, Model M, int accio, String titol) {
39         super(GUI, titol, true);
40         this.M = M;
41         this.accio = accio;
42         this.setIconImage(new ImageIcon("res/icon.png").getImage());
43         initComponents();
44         setLocationRelativeTo(null);
45         M.addObserver(this);
46         initValors();
47     }
48
49     /**
50      * Comprova que els camps hagin estat omplerts correctament
51      */
52     private boolean comprovarCamps(){

```



```

53     if (CaixaNom.getText().isEmpty()){
54         mostrarError(LANG.TXT_ERROR_NOM[M.getIdioma()]);
55         return false;
56     }
57     if (CaixaCognoms.getText().isEmpty()){
58         mostrarError(LANG.TXT_ERROR_COGNOMS[M.getIdioma()]);
59         return false;
60     }
61     try {
62         SimpleDateFormat SDF = new SimpleDateFormat("dd/MM/yyyy");
63         Date D = SDF.parse(CaixaDataNaixementDia.getText()+"/"+CaixaDataNaixementMes
        .getText()+"/"+CaixaDataNaixementAny.getText());
64         if (!SDF.format(D).equals((CaixaDataNaixementDia.getText()+"/"+
        CaixaDataNaixementMes.getText()+"/"+CaixaDataNaixementAny.getText()))){
65             mostrarError(LANG.TXT_ERROR_DATANAIXEMENT[M.getIdioma()]);
66             return false;
67         }
68     } catch (Exception e) {
69         mostrarError(LANG.TXT_ERROR_DATANAIXEMENT[M.getIdioma()]);
70         return false;
71     }
72     if (CaixaCodiHistorial.getText().length()!=12){
73         mostrarError(LANG.TXT_ERROR_CODIHISTORIAL[M.getIdioma()]);
74         return false;
75     }
76     if (!M.getPacientActual().getCodiHistorial().equals(CaixaCodiHistorial.getText()
        .toUpperCase()) && M.getPacient(CaixaCodiHistorial.getText())!=null){
77         mostrarError(LANG.TXT_ERROR_CODIHISTORIALEXISTEIX[M.getIdioma()]);
78         return false;
79     }
80     return true;
81 }
82
83 /**
84  * Mostra un missatge d'error per pantalla
85  * @param textError    Text del missatge d'error
86  */
87 private void mostrarError(String textError){
88     JOptionPane.showMessageDialog(this, textError, "Error", JOptionPane.
        ERROR_MESSAGE);
89 }
90
91 @SuppressWarnings("unchecked")
92 // <editor-fold defaultstate="collapsed" desc="Generated Code">
93 private void initComponents() {
94
95     LabelText = new javax.swing.JLabel();
96     LabelNom = new javax.swing.JLabel();
97     LabelCognoms = new javax.swing.JLabel();
98     LabelDataNaixement = new javax.swing.JLabel();
99     LabelSexe = new javax.swing.JLabel();
100    CaixaNom = new javax.swing.JTextField();
101    CaixaCognoms = new javax.swing.JTextField();
102    BotoAltaModificar = new javax.swing.JButton();
103    CaixaDataNaixementDia = new javax.swing.JTextField();
104    CaixaDataNaixementMes = new javax.swing.JTextField();
105    CaixaDataNaixementAny = new javax.swing.JTextField();
106    LabelFormatDataNaixement = new javax.swing.JLabel();
107    SelectorSexe = new java.awt.Choice();
108    LabelCodiHistorial = new javax.swing.JLabel();
109    CaixaCodiHistorial = new javax.swing.JTextField();
110
111    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
112    setResizable(false);
113
114    LabelText.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
115    LabelText.setText("Introduïu les dades del nou pacient:");
116
117    LabelNom.setText("Nom");
118

```

```

119     LabelCognoms.setText("Cognoms");
120
121     LabelDataNaixement.setText("Data de Naixement");
122
123     LabelSexe.setText("Sexe");
124
125     BotoAltaModificar.setText("Donar d'alta");
126     BotoAltaModificar.addActionListener(new java.awt.event.ActionListener() {
127         public void actionPerformed(java.awt.event.ActionEvent evt) {
128             BotoAltaModificarActionPerformed(evt);
129         }
130     });
131
132     CaixaDataNaixementDia.setText("dd");
133
134     CaixaDataNaixementMes.setText("mm");
135
136     CaixaDataNaixementAny.setText("yyyy");
137
138     LabelFormatDataNaixement.setText("(dd/mm/yyyy)");
139
140     SelectorSexe.addItem(LANG.TXT_SEXETXT[M.getIdioma()][0]);
141     SelectorSexe.addItem(LANG.TXT_SEXETXT[M.getIdioma()][1]);
142
143     LabelCodiHistorial.setText("Codi d'historial");
144
145     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
146     getContentPane().setLayout(layout);
147     layout.setHorizontalGroup(
148         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
149         .addGroup(layout.createSequentialGroup()
150             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
151                 .addGroup(layout.createSequentialGroup()
152                     .addContainerGap()
153                     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
154                         .addComponent(LabelText)
155                         .addGroup(layout.createSequentialGroup()
156                             .addGroup(layout.createParallelGroup(javax.swing.
157                                 GroupLayout.Alignment.LEADING)
158                                 .addComponent(LabelNom)
159                                 .addComponent(LabelCognoms)
160                                 .addComponent(LabelDataNaixement)
161                                 .addComponent(LabelSexe))
162                             .addGap(27, 27, 27)
163                             .addGroup(layout.createParallelGroup(javax.swing.
164                                 GroupLayout.Alignment.LEADING)
165                                 .addComponent(CaixaCognoms, javax.swing.GroupLayout.
166                                     PREFERRED_SIZE, 239, javax.swing.GroupLayout.
167                                     PREFERRED_SIZE)
168                                 .addComponent(CaixaNom, javax.swing.GroupLayout.
169                                     PREFERRED_SIZE, 118, javax.swing.GroupLayout.
170                                     PREFERRED_SIZE)
171                                 .addGroup(layout.createSequentialGroup()
172                                     .addGroup(layout.createParallelGroup(javax.swing.
173                                         GroupLayout.Alignment.TRAILING, false)
174                                         .addComponent(SelectorSexe, javax.swing.
175                                             GroupLayout.Alignment.LEADING, javax.
176                                                 swing.GroupLayout.DEFAULT_SIZE, javax.
177                                                 swing.GroupLayout.DEFAULT_SIZE, Short.
178                                                 MAX_VALUE)
179                                         .addGroup(javax.swing.GroupLayout.Alignment.
180                                             LEADING, layout.createSequentialGroup()
181                                                 .addComponent(CaixaDataNaixementDia,
182                                                     javax.swing.GroupLayout.
183                                                         PREFERRED_SIZE, 29, javax.swing.
184                                                         GroupLayout.PREFERRED_SIZE)
185                                                 .addPreferredGap(javax.swing.LayoutStyle.
186                                                     ComponentPlacement.RELATED)

```

```

171         .addComponent(CaixaDataNaixementMes,
172                        javax.swing.GroupLayout.
173                           PREFERRED_SIZE, 30, javax.swing.
174                              GroupLayout.PREFERRED_SIZE)
175         .addPreferredGap(javax.swing.LayoutStyle.
176                        .ComponentPlacement.RELATED)
177         .addComponent(CaixaDataNaixementAny,
178                        javax.swing.GroupLayout.
179                           PREFERRED_SIZE, 46, javax.swing.
180                              GroupLayout.PREFERRED_SIZE))
181         .addComponent(CaixaCodiHistorial, javax.
182                        swing.GroupLayout.Alignment.LEADING))
183         .addPreferredGap(javax.swing.LayoutStyle.
184                        .ComponentPlacement.UNRELATED)
185         .addComponent(LabelFormatDataNaixement)
186         .addGap(44, 44, 44))))))
187     .addGroup(layout.createSequentialGroup())
188     .addContainerGap()
189     .addComponent(LabelCodiHistorial))
190     .addGroup(layout.createSequentialGroup())
191     .addGap(143, 143, 143)
192     .addComponent(BotoAltaModificar)))
193     .addContainerGap(18, Short.MAX_VALUE))
194 );
195 layout.setVerticalGroup(
196     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
197     .addGroup(layout.createSequentialGroup())
198     .addContainerGap()
199     .addComponent(LabelText)
200     .addGap(18, 18, 18)
201     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
202     BASELINE)
203     .addComponent(LabelNom)
204     .addComponent(CaixaNom, javax.swing.GroupLayout.PREFERRED_SIZE,
205                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.
206                    PREFERRED_SIZE))
207     .addGap(4, 4, 4)
208     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
209     BASELINE)
210     .addComponent(LabelCognoms)
211     .addComponent(CaixaCognoms, javax.swing.GroupLayout.PREFERRED_SIZE,
212                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.
213                    PREFERRED_SIZE))
214     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
215     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
216     BASELINE)
217     .addComponent(CaixaDataNaixementDia, javax.swing.GroupLayout.
218                    PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.
219                    swing.GroupLayout.PREFERRED_SIZE)
220     .addComponent(CaixaDataNaixementMes, javax.swing.GroupLayout.
221                    PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.
222                    swing.GroupLayout.PREFERRED_SIZE)
223     .addComponent(CaixaDataNaixementAny, javax.swing.GroupLayout.
224                    PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.
225                    swing.GroupLayout.PREFERRED_SIZE)
226     .addComponent(LabelDataNaixement)
227     .addComponent(LabelFormatDataNaixement))
228     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
229     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
230     LEADING)
231     .addComponent(LabelSexe)
232     .addComponent(SelectorSexe, javax.swing.GroupLayout.PREFERRED_SIZE,
233                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.
234                    PREFERRED_SIZE))
235     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
236     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
237     BASELINE)
238     .addComponent(LabelCodiHistorial)
239     .addComponent(CaixaCodiHistorial, javax.swing.GroupLayout.
240                    PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.

```

```

214         swing.GroupLayout.PREFERRED_SIZE))
215         .addGap(18, 18, 18)
216         .addComponent(BotoAltaModificar)
217         .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE));
218
219     pack();
220 }// </editor-fold>
221
222 /**
223  * Conté les accions a dur a terme en polsar el botó
224  * @param evt    Event generat sobre el botó
225  */
226 private void BotoAltaModificarActionPerformed(java.awt.event.ActionEvent evt) {
227     SimpleDateFormat SDF;
228     Pacient P;
229     if (comprovarCamps()){
230         SDF = new SimpleDateFormat("dd/MM/yyyy");
231         Date D = null;
232         try {
233             D = SDF.parse(CaixaDataNaixementDia.getText()+"/"+CaixaDataNaixementMes.
234                             getText()+"/"+CaixaDataNaixementAny.getText());
235         } catch (Exception e) {}
236         P = new Pacient(CaixaCodiHistorial.getText().toUpperCase(),CaixaNom.getText
237                         (),CaixaCognoms.getText(),D,SelectorSexe.getSelectedIndex(),Utils.
238                             obtenirDataActual(),Utils.obtenirDataActual());
239         switch(accio){
240             case (CONST.PACIENT_CREAR):
241                 M.setPacient(P);
242                 break;
243             case (CONST.PACIENT_MODIFICAR):
244                 M.updPacient(P, M.getPacientActual().getCodiHistorial());
245                 break;
246         }
247         M.setPacientActual(P);
248         dispose();
249     }
250 }
251
252 // Variables declaration - do not modify
253 private javax.swing.JButton BotoAltaModificar;
254 private javax.swing.JTextField CaixaCodiHistorial;
255 private javax.swing.JTextField CaixaCognoms;
256 private javax.swing.JTextField CaixaDataNaixementAny;
257 private javax.swing.JTextField CaixaDataNaixementDia;
258 private javax.swing.JTextField CaixaDataNaixementMes;
259 private javax.swing.JTextField CaixaNom;
260 private javax.swing.JLabel LabelCodiHistorial;
261 private javax.swing.JLabel LabelCognoms;
262 private javax.swing.JLabel LabelDataNaixement;
263 private javax.swing.JLabel LabelFormatDataNaixement;
264 private javax.swing.JLabel LabelNom;
265 private javax.swing.JLabel LabelSexe;
266 private javax.swing.JLabel LabelText;
267 private java.awt.Choice SelectorSexe;
268 // End of variables declaration
269
270 /**
271  * Inicialitza les variables del panell
272  */
273 public void initValors(){
274     LabelCodiHistorial.setText(LANG.TXT_CODIHISTORIAL[M.getIdioma()]);
275     LabelNom.setText(LANG.TXT_NOM[M.getIdioma()]);
276     LabelCognoms.setText(LANG.TXT_COGNOMS[M.getIdioma()]);
277     LabelDataNaixement.setText(LANG.TXT_DATANAIXEMENT[M.getIdioma()]);
278     LabelSexe.setText(LANG.TXT_SEXE[M.getIdioma()]);
279     if (accio == CONST.PACIENT_CREAR){
280         LabelText.setText(LANG.TXT_DIALOGPACIENT1[M.getIdioma()]);
281         BotoAltaModificar.setText(LANG.TXT_DONARDALTA[M.getIdioma()]);
282         CaixaCodiHistorial.setText("");
283     }
284 }

```

```

280         CaixaCodiHistorial.enable();
281         CaixaNom.setText("");
282         CaixaCognoms.setText("");
283         CaixaDataNaixementDia.setText("dd");
284         CaixaDataNaixementMes.setText("mm");
285         CaixaDataNaixementAny.setText("yyyy");
286     } else {
287         LabelText.setText(LANG.TXT_DIALOGPACIENT2[M.getIdioma()]);
288         BotoAltaModificar.setText(LANG.TXT_MODIFICARDADES[M.getIdioma()]);
289         CaixaCodiHistorial.setText(M.getPacientActual().getCodiHistorial());
290         CaixaCodiHistorial.disable();
291         CaixaNom.setText(M.getPacientActual().getNom());
292         CaixaCognoms.setText(M.getPacientActual().getCognoms());
293         CaixaDataNaixementDia.setText(Utils.obtenirDataTXT(M.getPacientActual().
294             getDataNaixement(), CONST.DATA_DIA, M.getIdioma()));
295         CaixaDataNaixementMes.setText(Utils.obtenirDataTXT(M.getPacientActual().
296             getDataNaixement(), CONST.DATA_MES, M.getIdioma()));
297         CaixaDataNaixementAny.setText(Utils.obtenirDataTXT(M.getPacientActual().
298             getDataNaixement(), CONST.DATA_ANY, M.getIdioma()));
299         SelectorSexe.select(M.getPacientActual().getSexe());
300     }
301 }
302
303 /**
304  * Actualitza la vista amb les noves dades del Model (patró MVC)
305  * @param o      Observable que notifica el canvi en el Model
306  * @param arg    Arguments
307  */
308 public void update(Observable o, Object arg) {}
309
310 }

```

## Codi D.16: (AppSeguiment) GUI.java

```

1 package Vista;
2
3
4 import Utils.Grafics;
5 import Constants.CONST;
6 import Constants.LANG;
7 import Dades.Alarma;
8 import Dades.Model;
9 import Dades.Pacient;
10 import Main.App;
11 import Utils.Utils;
12 import Utils.UtilsVista;
13 import java.awt.Color;
14 import java.awt.Component;
15 import java.awt.Desktop;
16 import java.awt.FileDialog;
17 import java.awt.Frame;
18 import java.awt.event.ActionEvent;
19 import java.awt.event.ActionListener;
20 import java.awt.image.BufferedImage;
21 import java.io.IOException;
22 import java.net.URI;
23 import java.net.URISyntaxException;
24 import java.util.Date;
25 import java.util.Iterator;
26 import java.util.List;
27 import java.util.Observable;
28 import java.util.logging.Level;
29 import java.util.logging.Logger;
30 import javax.swing.ButtonGroup;
31 import javax.swing.ImageIcon;
32 import javax.swing.JDialog;
33 import javax.swing.JFrame;
34 import javax.swing.JMenu;
35 import javax.swing.JMenuBar;

```

```

36 import javax.swing.JMenuItem;
37 import javax.swing.JOptionPane;
38 import javax.swing.JRadioButtonMenuItem;
39 import javax.swing.JTable;
40 import javax.swing.table.AbstractTableModel;
41 import javax.swing.table.DefaultTableModel;
42 import javax.swing.table.TableCellRenderer;
43
44 /**
45  * Classe que conté tota la gestió de la GUI (Graphic User Interface). Actua com a Vista
46  * dins del patró MVC (Model-View-Controller)
47  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
48  * @version 1.0 (Juny de 2010)
49  */
50 public class GUI extends JFrame implements Vista {
51
52     /** Referència a l'App */
53     private App A;
54     /** Referència al model de dades */
55     private Model M;
56
57     /** Element gràfic de la GUI: barra de menú */
58     private JMenuBar MB;
59     /** Element gràfic de la GUI: agrupació de botons */
60     private ButtonGroup BG;
61     /** Element gràfic de la GUI: menú arxiu */
62     private JMenu MenuArxiu;
63     /** Element gràfic de la GUI: menú idioma */
64     private JMenu MenuIdioma;
65     /** Element gràfic de la GUI: menú ajuda */
66     private JMenu MenuAjuda;
67     /** Element gràfic de la GUI: menú ítem nou pacient */
68     private JMenuItem MenuItemNouPacient;
69     /** Element gràfic de la GUI: menú ítem modificar pacient */
70     private JMenuItem MenuItemModificarPacient;
71     /** Element gràfic de la GUI: menú ítem eliminar pacient */
72     private JMenuItem MenuItemEliminarPacient;
73     /** Element gràfic de la GUI: menú ítem obtenir historial */
74     private JMenuItem MenuItemObtenirHistorial;
75     /** Element gràfic de la GUI: menú ítem generar informe PDF */
76     private JMenuItem MenuItemGenerarInformePDF;
77     /** Element gràfic de la GUI: menú ítem sortir */
78     private JMenuItem MenuItemSortir;
79     /** Element gràfic de la GUI: menú ítem mostrar ajuda */
80     private JMenuItem MenuItemMostrarAjuda;
81     /** Element gràfic de la GUI: menú ítem idioma català */
82     private JRadioButtonMenuItem MenuItemIdiomaCatala;
83     /** Element gràfic de la GUI: menú ítem idioma castellà */
84     private JRadioButtonMenuItem MenuItemIdiomaCastella;
85     /** Element gràfic de la GUI: menú ítem idioma anglès */
86     private JRadioButtonMenuItem MenuItemIdiomaAngles;
87     /** Element gràfic de la GUI: */
88     private JDialog DialogPacient;
89
90     /**
91     * Crea una nova instància de GUI
92     * @param A Referència a l'aplicació
93     * @param M Referència al Model de dades
94     */
95     public GUI(App A, Model M) {
96         this.A = A;
97         this.M = M;
98         setIconImage(new ImageIcon("res/icon.png").getImage());
99         setTitle("PPStop Seguiment");
100         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
101         initComponents();
102         crearMenu();
103         setLocationRelativeTo(null);
104         M.addObserver(this);
105         pack();

```

```

105     actualitzaParametres();
106 }
107
108 /** This method is called from within the constructor to
109  * initialize the form.
110  * WARNING: Do NOT modify this code. The content of this method is
111  * always regenerated by the Form Editor.
112  */
113 @SuppressWarnings("unchecked")
114 // <editor-fold defaultstate="collapsed" desc="Generated Code">
115 private void initComponents() {
116
117     jScrollPane1 = new javax.swing.JScrollPane();
118     TaulaAlarmes = new javax.swing.JTable(){
119         public Component prepareRenderer(TableCellRenderer renderer, int row,
120             int col) {
121             Component comp = super.prepareRenderer(renderer, row, col);
122             if (row%2 == 0) {
123                 comp.setBackground(Color.WHITE);
124             } else {
125                 comp.setBackground(new Color(240,245,255));
126             }
127             if (col==1){
128                 if (this.getModel().getValueAt(row, col).toString().equals(" D")){
129                     comp.setForeground(Color.BLUE);
130                 } else {
131                     comp.setForeground(Color.RED);
132                 }
133             } else {
134                 comp.setForeground(Color.BLACK);
135             }
136             return comp;
137         }
138     };
139     Grafic1 = new javax.swing.JLabel();
140     SelectorGrafic = new java.awt.Choice();
141     LabelSeleccionarPacient = new javax.swing.JLabel();
142     jSeparator1 = new javax.swing.JSeparator();
143     SelectorPacient = new java.awt.Choice();
144     LabelSeleccionarGrafic = new javax.swing.JLabel();
145     jPanel1 = new javax.swing.JPanel();
146     LabelDataAlta = new javax.swing.JLabel();
147     LabelEdat = new javax.swing.JLabel();
148     LabelSexe = new javax.swing.JLabel();
149     LabelDarreraTransferencia = new javax.swing.JLabel();
150     LabelCodiHistorial = new javax.swing.JLabel();
151     LabelCodiHistorialAns = new javax.swing.JLabel();
152     LabelEdatAns = new javax.swing.JLabel();
153     LabelSexeAns = new javax.swing.JLabel();
154     LabelDataAltaAns = new javax.swing.JLabel();
155     LabelDarreraTransferenciaAns = new javax.swing.JLabel();
156
157     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
158     setResizable(false);
159
160     TaulaAlarmes.setModel(new javax.swing.table.DefaultTableModel(
161
162     ));
163     TaulaAlarmes.setEnabled(false);
164     TaulaAlarmes.setGridColor(new java.awt.Color(204, 204, 204));
165     jScrollPane1.setViewportView(TaulaAlarmes);
166
167     SelectorGrafic.addItemListener(new java.awt.event.ItemListener() {
168         public void itemStateChanged(java.awt.event.ItemEvent evt) {
169             SelectorGraficListener(evt);
170         }
171     });
172
173     LabelSeleccionarPacient.setText("Selecciona el pacient a analitzar:");

```



```

174 SelectorPacient.addItemListener(new java.awt.event.ItemListener() {
175     public void itemStateChanged(java.awt.event.ItemEvent evt) {
176         SelectorPacientListener(evt);
177     }
178 });
179
180 LabelSeleccionarGrafic.setText("Selecciona el tipus de gràfic:");
181
182 JPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(""));
183
184 LabelDataAlta.setText("Data d'alta:");
185
186 LabelEdat.setText("Edat:");
187
188 LabelSexe.setText("Sexe:");
189
190 LabelDarreraTransferencia.setText("Darrera transferència:");
191
192 LabelCodiHistorial.setText("Codi historial:");
193
194 LabelCodiHistorialAns.setForeground(new java.awt.Color(102, 153, 255));
195 LabelCodiHistorialAns.setText("--");
196
197 LabelEdatAns.setForeground(new java.awt.Color(102, 153, 255));
198 LabelEdatAns.setText("--");
199
200 LabelSexeAns.setForeground(new java.awt.Color(102, 153, 255));
201 LabelSexeAns.setText("--");
202
203 LabelDataAltaAns.setForeground(new java.awt.Color(102, 153, 255));
204 LabelDataAltaAns.setText("--");
205
206 LabelDarreraTransferenciaAns.setForeground(new java.awt.Color(102, 153, 255));
207 LabelDarreraTransferenciaAns.setText("--");
208
209 javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
210 jPanel1.setLayout(jPanel1Layout);
211 jPanel1Layout.setHorizontalGroup(
212     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
213     .addGroup(jPanel1Layout.createSequentialGroup()
214         .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
215             .add(jPanel1Layout.createSequentialGroup()
216                 .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
217                     .add(LabelCodiHistorial, javax.swing.GroupLayout.PREFERRED_SIZE, 87, javax.swing.GroupLayout.PREFERRED_SIZE)
218                     .add(LabelDataAlta, javax.swing.GroupLayout.PREFERRED_SIZE, 75, javax.swing.GroupLayout.PREFERRED_SIZE))
219                 .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
220                     .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
221                         .add(LabelDataAltaAns, javax.swing.GroupLayout.PREFERRED_SIZE, 132, javax.swing.GroupLayout.PREFERRED_SIZE)
222                         .add(LabelCodiHistorialAns, javax.swing.GroupLayout.PREFERRED_SIZE, 128, javax.swing.GroupLayout.PREFERRED_SIZE))
223                     .add(jPanel1Layout.createSequentialGroup()
224                         .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
225                             .add(LabelEdat, javax.swing.GroupLayout.PREFERRED_SIZE, 18, javax.swing.GroupLayout.PREFERRED_SIZE)
226                             .add(LabelEdatAns, javax.swing.GroupLayout.PREFERRED_SIZE, 92, javax.swing.GroupLayout.PREFERRED_SIZE)
227                             .add(LabelSexe, javax.swing.GroupLayout.PREFERRED_SIZE, 18, javax.swing.GroupLayout.PREFERRED_SIZE)
228                             .add(LabelSexeAns, javax.swing.GroupLayout.PREFERRED_SIZE, 73, javax.swing.GroupLayout.PREFERRED_SIZE))
229                         .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
230                             .add(LabelDarreraTransferencia, javax.swing.GroupLayout.PREFERRED_SIZE, 136, javax.swing.GroupLayout.PREFERRED_SIZE)

```



```

    )
234     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
        UNRELATED)
235     .addComponent(LabelDarreraTransferenciaAns, javax.swing.
        GroupLayout.PREFERRED_SIZE, 130, javax.swing.GroupLayout.
        PREFERRED_SIZE)))
236     .addContainerGap(40, Short.MAX_VALUE))
237 );
238 jPanel1Layout.setVerticalGroup(
239     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
240     .addGroup(jPanel1Layout.createSequentialGroup()
241         .addContainerGap()
242         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.BASELINE)
243             .addComponent(LabelCodiHistorial)
244             .addComponent(LabelCodiHistorialAns)
245             .addComponent(LabelEdat)
246             .addComponent(LabelEdatAns)
247             .addComponent(LabelSexe)
248             .addComponent(LabelSexeAns))
249         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 13,
            Short.MAX_VALUE)
250         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.BASELINE)
251             .addComponent(LabelDataAlta)
252             .addComponent(LabelDataAltaAns)
253             .addComponent(LabelDarreraTransferencia)
254             .addComponent(LabelDarreraTransferenciaAns))
255         .addContainerGap())
256 );
257
258 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
259 getContentPane().setLayout(layout);
260 layout.setHorizontalGroup(
261     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
262     .addGroup(layout.createSequentialGroup()
263         .addContainerGap()
264         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
            LEADING)
265             .addComponent(jSeparator1, javax.swing.GroupLayout.DEFAULT_SIZE,
                996, Short.MAX_VALUE)
266             .addGroup(layout.createSequentialGroup()
267                 .addComponent(jScrollPane1, javax.swing.GroupLayout.
                    PREFERRED_SIZE, 392, javax.swing.GroupLayout.PREFERRED_SIZE)
268                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
                    RELATED)
269                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
                    Alignment.LEADING)
270                     .addComponent(Grafic1, javax.swing.GroupLayout.DEFAULT_SIZE,
                        598, Short.MAX_VALUE)
271                     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
                        Alignment.TRAILING, false)
272                         .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
                            layout.createSequentialGroup()
273                             .addGap(9, 9, 9)
274                             .addComponent(LabelSeleccionarGrafic)
275                             .addPreferredGap(javax.swing.LayoutStyle.
                                ComponentPlacement.RELATED)
276                             .addComponent(SelectorGrafic, javax.swing.
                                    GroupLayout.DEFAULT_SIZE, javax.swing.
                                        GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
277                         .addComponent(jPanel1, javax.swing.GroupLayout.Alignment.
                            LEADING, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.
                                    GroupLayout.PREFERRED_SIZE))))))
278     .addGroup(layout.createSequentialGroup()
279         .addComponent(LabelSeleccionarPacient)
280         .addGap(26, 26, 26)
281         .addComponent(SelectorPacient, javax.swing.GroupLayout.
            PREFERRED_SIZE, 337, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

282         ))
283         .addContainerGap();
284     layout.setVerticalGroup(
285         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
286         .addGroup(layout.createSequentialGroup()
287             .addContainerGap()
288             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
289                 LEADING)
289                 .addComponent(LabelSeleccionarPacient)
290                 .addComponent(SelectorPacient, javax.swing.GroupLayout.
291                     PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.
292                         swing.GroupLayout.PREFERRED_SIZE))
291             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
292             .addComponent(jSeparator1, javax.swing.GroupLayout.DEFAULT_SIZE, 1,
293                 Short.MAX_VALUE)
293             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
294             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
295                 TRAILING)
295                 .addGroup(layout.createSequentialGroup()
296                     .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
297                         javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
297                     .addGap(18, 18, 18)
298                     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
299                         Alignment.LEADING)
300                         .addComponent(LabelSeleccionarGrafic)
301                         .addComponent(SelectorGrafic, javax.swing.GroupLayout.
302                             PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
303                                 javax.swing.GroupLayout.PREFERRED_SIZE))
303                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
304                         RELATED)
304                     .addComponent(Grafic1, javax.swing.GroupLayout.PREFERRED_SIZE,
305                         311, javax.swing.GroupLayout.PREFERRED_SIZE))
305                     .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
306                         434, Short.MAX_VALUE))
306                 .addContainerGap())
307             );
308     pack();
309 } // </editor-fold>
310 private void SelectorGraficListener(java.awt.event.ItemEvent evt) {
311     actualitzaGrafic();
312 }
313
314 private void SelectorPacientListener(java.awt.event.ItemEvent evt) {
315     M.setPacientActual(M.getPacients().get(SelectorPacient.getSelectedIndex()));
316 }
317
318 /**
319  * Crea el menú de l'aplicació
320  */
321 private void crearMenu(){
322     MB = new JMenuBar();
323     setJMenuBar(MB);
324     MenuArxiu = new JMenu(LANG.MENU_ARXIU[M.getIdioma()]);
325     MenuIdioma = new JMenu(LANG.MENU_IDIOMA[M.getIdioma()]);
326     MenuAjuda = new JMenu(LANG.MENU_AJUDA[M.getIdioma()]);
327     MB.add(MenuArxiu);
328     MB.add(MenuIdioma);
329     MB.add(MenuAjuda);
330     MenuItemNouPacient = new JMenuItem(LANG.MENU_NOUPACIENT[M.getIdioma()]);
331     MenuItemNouPacient.setIcon(new ImageIcon("res/crear.png"));
332     MenuItemModificarPacient = new JMenuItem(LANG.MENU_MODIFICARPACIENT[M.getIdioma()]);
333     MenuItemModificarPacient.setIcon(new ImageIcon("res/modificar.png"));
334     MenuItemEliminarPacient = new JMenuItem(LANG.MENU_ELIMINARPACIENT[M.getIdioma()]);
335     MenuItemEliminarPacient.setIcon(new ImageIcon("res/eliminar.png"));
336     MenuItemObtenirHistorial = new JMenuItem(LANG.MENU_OBTENIRHISTORIAL[M.getIdioma()]);

```

```

    ());
337 MenuItemObtenirHistorial.setIcon(new ImageIcon("res/obtenir.png"));
338 MenuItemGenerarInformePDF = new JMenuItem(LANG.MENU_GENERARINFORMEPDF[M.
    getId idioma()));
339 MenuItemGenerarInformePDF.setIcon(new ImageIcon("res/informe.png"));
340 MenuItemSortir = new JMenuItem(LANG.MENU_SORTIR[M.getId idioma()));
341 MenuItemSortir.setIcon(new ImageIcon("res/sortir.png"));
342 MenuItemMostrarAjuda = new JMenuItem(LANG.MENU_MOSTRARAJUDA[M.getId idioma()));
343 MenuItemMostrarAjuda.setIcon(new ImageIcon("res/ajuda.png"));
344 MenuItemIdiomaCatala = new JRadioButtonMenuItem(LANG.MENU_CATALA[M.getId idioma()));
    ;
345 MenuItemIdiomaCastella = new JRadioButtonMenuItem(LANG.MENU_CASTELLA[M.getId idioma
    (]]);
346 MenuItemIdiomaAngles = new JRadioButtonMenuItem(LANG.MENU_ANGLES[M.getId idioma()));
    ;
347 MenuItemIdiomaCatala.setIcon(new ImageIcon("res/ca.png"));
348 MenuItemIdiomaCastella.setIcon(new ImageIcon("res/es.png"));
349 MenuItemIdiomaAngles.setIcon(new ImageIcon("res/en.png"));
350 BG = new ButtonGroup();
351 BG.add(MenuItemIdiomaCatala);
352 BG.add(MenuItemIdiomaCastella);
353 BG.add(MenuItemIdiomaAngles);
354 MenuArxiu.add(MenuItemNouPacient);
355 MenuArxiu.add(MenuItemModificarPacient);
356 MenuArxiu.add(MenuItemEliminarPacient);
357 MenuArxiu.addSeparator();
358 MenuArxiu.add(MenuItemObtenirHistorial);
359 MenuArxiu.add(MenuItemGenerarInformePDF);
360 MenuArxiu.addSeparator();
361 MenuArxiu.add(MenuItemSortir);
362 MenuIdioma.add(MenuItemIdiomaCatala);
363 MenuIdioma.add(MenuItemIdiomaCastella);
364 MenuIdioma.add(MenuItemIdiomaAngles);
365 MenuAjuda.add(MenuItemMostrarAjuda);
366 switch(M.getId idioma()){
367     case CONST.IDIOMA_CA:
368         BG.setSelected(MenuItemIdiomaCatala.getModel(),true);
369         break;
370     case CONST.IDIOMA_ES:
371         BG.setSelected(MenuItemIdiomaCastella.getModel(),true);
372         break;
373     case CONST.IDIOMA_EN:
374         BG.setSelected(MenuItemIdiomaAngles.getModel(),true);
375         break;
376 }
377 MenuItemNouPacient.addActionListener(new ActionListener() {
378     public void actionPerformed(ActionEvent evt) {
379         mostrarDialogPacient(CONST.PACIENT_CREAR,LANG.MENU_NOUPACIENT[M.
            getId idioma()));
380     }
381 });
382 MenuItemModificarPacient.addActionListener(new ActionListener() {
383     public void actionPerformed(ActionEvent evt) {
384         if (M.getPacientActual()==null){
385             mostrarAlerta(LANG.TXT_ERROR_NOPACIENT[M.getId idioma()],JOptionPane.
                ERROR_MESSAGE);
386         } else {
387             mostrarDialogPacient(CONST.PACIENT_MODIFICAR,LANG.
                MENU_MODIFICARPACIENT[M.getId idioma()));
388         }
389     }
390 });
391 MenuItemEliminarPacient.addActionListener(new ActionListener() {
392     public void actionPerformed(ActionEvent evt) {
393         if(M.getPacientActual()==null){
394             mostrarAlerta(LANG.TXT_ERROR_NOPACIENT[M.getId idioma()],JOptionPane.
                ERROR_MESSAGE);
395         } else {
396             int resp = mostrarConfirmacio(Utils.substituirStr(LANG.
                TXT_CONFIRMACIO_ELIMINAR[M.getId idioma()], "{X}", M.

```

```

        getPacientActual().getCognoms().toUpperCase()+"", "+M.
        getPacientActual().getNom().toUpperCase());
397     if (resp == JOptionPane.YES_OPTION){
398         M.delPacient(M.getPacientActual());
399         M.setPacientActual(null);
400     }
401 }
402 }
403 });
404 MenuItemObtenirHistorial.addActionListener(new ActionListener() {
405     public void actionPerformed(ActionEvent evt) {
406         A.obtenirHistorial(M.getPacientActual());
407     }
408 });
409 MenuItemGenerarInformePDF.addActionListener(new ActionListener() {
410     public void actionPerformed(ActionEvent evt) {
411         List<Alarma> Alarmes = M.getAlarmes(M.getPacientActual(),null);
412         if (Alarmes.size()>0){
413             String path = seleccionaPath("informe");
414             if (path!=null){
415                 try {
416                     A.generarInforme(M.getPacientActual(), path);
417                 } catch (Exception ex) {
418                     mostrarAlerta(LANG.TXT_ERROR_GENERACIOINFORME[M.getIdioma()
419                                     ],JOptionPane.ERROR_MESSAGE);
420                 }
421             } else {
422                 mostrarAlerta(LANG.TXT_ERROR_NOALARMES[M.getIdioma()],JOptionPane.
423                     ERROR_MESSAGE);
424             }
425         }
426     });
427 MenuItemIdiomaCatala.addActionListener(new ActionListener() {
428     public void actionPerformed(ActionEvent arg0) {
429         M.setIdioma(CONST.IDIOMA_CA);
430     }
431 });
432 MenuItemIdiomaCastella.addActionListener(new ActionListener() {
433     public void actionPerformed(ActionEvent arg0) {
434         M.setIdioma(CONST.IDIOMA_ES);
435     }
436 });
437 MenuItemIdiomaAngles.addActionListener(new ActionListener() {
438     public void actionPerformed(ActionEvent arg0) {
439         M.setIdioma(CONST.IDIOMA_EN);
440     }
441 });
442 MenuItemMostrarAjuda.addActionListener(new ActionListener() {
443     public void actionPerformed(ActionEvent arg0) {
444         try {
445             URI uri = new URI("http://lewis.upc.es/ppstop/manuals/
446                 manual_ppstop_seguiment_v1.0.pdf");
447             Desktop.getDesktop().browse(uri);
448         } catch (IOException ex) {
449             Logger.getLogger(GUI.class.getName()).log(Level.SEVERE, null, ex);
450         } catch (URISyntaxException ex) {
451             Logger.getLogger(GUI.class.getName()).log(Level.SEVERE, null, ex);
452         }
453     }
454 });
455 MenuItemSortir.addActionListener(new ActionListener() {
456     public void actionPerformed(ActionEvent arg0) {
457         System.exit(0);
458     }
459 });
460 }
461
462 /**
463  * Mostra la finestra de creació/modificació d'un pacient

```

```

462     * @param accio      Accio a realitzar sobre el dialog (crear/modificar)
463     * @param titol      Títol del dialog
464     */
465     private void mostrarDialogPacient(int accio, String titol){
466         DialogPacient = new DialogPacient(this,M,accio,titol);
467         DialogPacient.setVisible(true);
468     }
469
470     /**
471     * Mostra un missatge de confirmació per pantalla
472     * @param textConfirmacio Text del missatge de confirmació
473     */
474     private int mostrarConfirmacio(String textConfirmacio){
475         return JOptionPane.showConfirmDialog(null, textConfirmacio, "PPStop Seguiment",
476             JOptionPane.YES_NO_OPTION);
477     }
478
479     /**
480     * Mostra una finestra selectora de ruta i nom de fitxer
481     * @param nomfitxer      Nom del fitxer per defecte
482     */
483     private String seleccionaPath(String nomfitxer){
484         FileDialog fd = new FileDialog(new Frame(),LANG.MENU_GENERARINFORMEPDF[M.
485             getIdioma()],FileDialog.SAVE);
486         fd.setFile(nomfitxer);
487         fd.setDirectory(".\\");
488         fd.setIconImage(new ImageIcon("res/icon.png").getImage());
489         fd.show();
490         return ((fd.getDirectory()==null) ? null : fd.getDirectory()+System.getProperty(
491             "file.separator")+fd.getFile());
492     }
493
494     /**
495     * Actualitza els paràmetres de la GUI
496     */
497     private void actualitzaParametres(){
498         MenuArxiu.setText(LANG.MENU_ARXIU[M.getIdioma()]);
499         MenuIdioma.setText(LANG.MENU_IDIOMA[M.getIdioma()]);
500         MenuAjuda.setText(LANG.MENU_AJUDA[M.getIdioma()]);
501         MenuItemNouPacient.setText(LANG.MENU_NOUPACIENT[M.getIdioma()]);
502         MenuItemModificarPacient.setText(LANG.MENU_MODIFICARPACIENT[M.getIdioma()]);
503         MenuItemEliminarPacient.setText(LANG.MENU_ELIMINARPACIENT[M.getIdioma()]);
504         MenuItemObtenirHistorial.setText(LANG.MENU_OBTENIRHISTORIAL[M.getIdioma()]);
505         MenuItemGenerarInformePDF.setText(LANG.MENU_GENERARINFORMEPDF[M.getIdioma()]);
506         MenuItemSortir.setText(LANG.MENU_SORTIR[M.getIdioma()]);
507         MenuItemMostrarAjuda.setText(LANG.MENU_MOSTRARAJUDA[M.getIdioma()]);
508         MenuItemIdiomaCatala.setText(LANG.MENU_CATALA[M.getIdioma()]);
509         MenuItemIdiomaCastella.setText(LANG.MENU_CASTELLA[M.getIdioma()]);
510         MenuItemIdiomaAngles.setText(LANG.MENU_ANGLES[M.getIdioma()]);
511         LabelSeleccionarGrafic.setText(LANG.TXT_SELECCIONARGRAFIC[M.getIdioma()]);
512         SelectorPacient.removeAll();
513         List<Pacient> Pacients = M.getPacients();
514         Iterator it = Pacients.iterator();
515         Pacient Pacient;
516         while (it.hasNext()){
517             Pacient = (Pacient)it.next();
518             SelectorPacient.addItem(Pacient.getCognoms().toUpperCase() + ", " + Pacient.
519                 getNom().toUpperCase());
520             if (M.getPacientActual()!=null && M.getPacientActual().getCodiHistorial().
521                 equals(Pacient.getCodiHistorial()))
522                 SelectorPacient.select(SelectorPacient.getItemCount()-1);
523         }
524         if (Pacients.size()>0 && M.getPacientActual()==null)
525             M.setPacientActual(Pacients.get(0));
526         LabelSeleccionarPacient.setText(LANG.TXT_SELECCIONARPACIENT[M.getIdioma()]);
527         if (M.getAlarmes(M.getPacientActual(),null).size()>0){
528             int SelectorGraficItem = SelectorGrafic.getSelectedIndex();
529             SelectorGrafic.removeAll();
530             SelectorGrafic.addItem(LANG.TXT_GRAFIC1[M.getIdioma()]);
531             SelectorGrafic.addItem(LANG.TXT_GRAFIC2[M.getIdioma()]);
532         }

```

```

527         SelectorGrafic.addItem(LANG.TXT_GRAFIC3[M.getIdioma()]);
528         SelectorGrafic.addItem(LANG.TXT_GRAFIC4[M.getIdioma()]);
529         if (SelectorGraficItem!=-1)
530             SelectorGrafic.select(SelectorGraficItem);
531         actualitzaTaula();
532         actualitzaGrafic();
533     } else {
534         TaulaAlarmes.setModel(new DefaultTableModel() {});
535         SelectorGrafic.removeAll();
536         Grafic1.setIcon(null);
537     }
538     if (M.getPacientActual()!=null)
539         actualitzaDadesPacient();
540     crearMenu();
541 }
542
543 /**
544  * Actualitza la taula d'alarmes de la GUI
545  */
546 private void actualitzaTaula(){
547     final List<Alarma> LA = M.getAlarmes(M.getPacientActual(),null);
548     TaulaAlarmes.setModel( new AbstractTableModel() {
549         protected String[] columnNames = {"N","T",LANG.TXT_ANY[M.getIdioma()],LANG.
550             TXT_MES[M.getIdioma()],LANG.TXT_DIA[M.getIdioma()],LANG.TXT_HORA[M.
551             getIdioma()], "Dif."};
552         @Override public String getColumnName(int col) { return columnNames[col].
553             toString(); }
554         public int getRowCount() { return M.getAlarmes(M.getPacientActual(), null).
555             size(); }
556         public int getColumnCount() { return columnNames.length; }
557         public Object getValueAt(int row, int col) {
558             Alarma Alarma = LA.get(row);
559             Date dataAnterior = (row==0) ? null : LA.get(row-1).getDataAlarma();
560             switch(col){
561                 case 0:
562                     return ("#" + (row+1));
563                 case 1:
564                     return (Alarma.getDataAlarma().getHours()>8 && Alarma.
565                         getDataAlarma().getHours()<22) ? " D " : " N ";
566                 case 2:
567                     return (" " + Utils.obtenirDataTXT(Alarma.getDataAlarma(),CONST.
568                         DATA_ANY,M.getIdioma()));
569                 case 3:
570                     return (" " + Utils.obtenirDataTXT(Alarma.getDataAlarma(),CONST.
571                         DATA_MESTXT,M.getIdioma()));
572                 case 4:
573                     return (" " + Utils.obtenirDataTXT(Alarma.getDataAlarma(),CONST.
574                         DATA_DIA,M.getIdioma()));
575                 case 5:
576                     return (" " + Utils.obtenirDataTXT(Alarma.getDataAlarma(),CONST.
577                         DATA_HORA,M.getIdioma()));
578                 case 6:
579                     return (" " + Utils.obtenirDiferenciaData(Alarma.getDataAlarma()
580                         ,dataAnterior,M.getIdioma()));
581                 default:
582                     return "--";
583             }
584         }
585     });
586     for (int i = 0; i < TaulaAlarmes.getColumnCount(); i++) {
587         switch(i){
588             case 0:
589                 TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((10*
590                     TaulaAlarmes.getWidth())/100);
591                 break;
592             case 1:
593                 TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((10*
594                     TaulaAlarmes.getWidth())/100);
595                 break;
596             case 2:

```

```

585         TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((15*
586             TaulaAlarmes.getWidth())/100);
587         break;
588     case 3:
589         TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((15*
590             TaulaAlarmes.getWidth())/100);
591         break;
592     case 4:
593         TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((15*
594             TaulaAlarmes.getWidth())/100);
595         break;
596     case 5:
597         TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((15*
598             TaulaAlarmes.getWidth())/100);
599         break;
600     case 6:
601         TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((20*
602             TaulaAlarmes.getWidth())/100);
603         break;
604     default:
605         TaulaAlarmes.getColumnModel().getColumn(i).setPreferredWidth((33*
606             TaulaAlarmes.getWidth())/100);
607         break;
608     }
609 }
610 }
611
612 /**
613  * Actualitza el gràfic d'alarmes de la GUI
614  */
615 private void actualitzaGrafic(){
616     Graphics G = new Graphics();
617     BufferedImage BI = null;
618     switch(SelectorGrafic.getSelectedIndex()){
619         case CONST.GRAFIC_EVOLUCIOMENSUAL:
620             BI = G.getGBarresEnters(M.getDadesGBarresMensual(M.getPacientActual()));
621             break;
622         case CONST.GRAFIC_DARRERS3ODIES:
623             BI = G.getGBarresEnters(M.getDadesGBarresUltimMes(M.getPacientActual()));
624             break;
625         case CONST.GRAFIC_DISTRIBUCIOHORARIA:
626             BI = G.getGBarresDecimals(M.getDadesGBarresPerHores(M.getPacientActual()));
627             break;
628         case CONST.GRAFIC_DISTRIBUCIODIURNANOCTURNA:
629             BI = G.getGFormatge(M.getDadesGFormatgeDiurnNocturn(M.getPacientActual()));
630             break;
631     }
632     Grafic1.setIcon(new ImageIcon(UtilsVista.EscalarImatge(BI,600,300)));
633 }
634
635 /**
636  * Actualitza les dades del pacient seleccionat a la GUI
637  */
638 private void actualitzaDadesPacient(){
639     LabelCodiHistorial.setText(LANG.TXT_CODIHISTORIAL[M.getIdioma()]+":");
640     LabelEdat.setText(LANG.TXT_EDAT[M.getIdioma()]+":");
641     LabelSexe.setText(LANG.TXT_SEXE[M.getIdioma()]+":");
642     LabelDataAlta.setText(LANG.TXT_DATAALTA[M.getIdioma()]+":");
643     LabelDarreraTransferencia.setText(LANG.TXT_DARRERATRANSFERENCIA[M.getIdioma()]+":");
644     LabelCodiHistorialAns.setText(M.getPacientActual().getCodiHistorial());
645     LabelEdatAns.setText(String.valueOf(Utils.obtenirEdat(M.getPacientActual().
646         getDataNaixement()))+" "+LANG.TXT_ANYS[M.getIdioma()]);
647     LabelSexeAns.setText(LANG.TXT_SEXETXT[M.getIdioma()][M.getPacientActual().
648         getSexe()]);
649     LabelDataAltaAns.setText(Utils.obtenirDataTXT(M.getPacientActual().getDataAlta(),
650         CONST.DATA_CURTA,M.getIdioma()));

```



```

642         LabelDarreraTransferenciaAns.setText(Utils.obtenirDataTXT(M.getPacientActual().
        getDarreraTransferencia(),CONST.DATA_SENCERA,M.getIdioma()));
643     }
644
645     /**
646     * Mostra un missatge d'alerta d'un tipus determinat (informació, error...)
647     * @param text    Missatge d'alerta
648     */
649     public void mostrarAlerta(String text, int tipus){
650         JOptionPane.showMessageDialog(this, text, "PPStop Seguiment", tipus);
651     }
652
653     /**
654     * Actualitza la vista amb les noves dades del Model (patró MVC)
655     * @param o        Observable que notifica el canvi en el Model
656     * @param arg      Arguments
657     */
658     public void update(Observable o, Object arg) {
659         actualitzaParametres();
660     }
661
662     // Variables declaration - do not modify
663     private javax.swing.JLabel Grafic1;
664     private javax.swing.JLabel LabelCodiHistorial;
665     private javax.swing.JLabel LabelCodiHistorialAns;
666     private javax.swing.JLabel LabelDarreraTransferencia;
667     private javax.swing.JLabel LabelDarreraTransferenciaAns;
668     private javax.swing.JLabel LabelDataAlta;
669     private javax.swing.JLabel LabelDataAltaAns;
670     private javax.swing.JLabel LabelEdat;
671     private javax.swing.JLabel LabelEdatAns;
672     private javax.swing.JLabel LabelSeleccionarGrafic;
673     private javax.swing.JLabel LabelSeleccionarPacient;
674     private javax.swing.JLabel LabelSexe;
675     private javax.swing.JLabel LabelSexeAns;
676     private java.awt.Choice SelectorGrafic;
677     private java.awt.Choice SelectorPacient;
678     private javax.swing.JTable TaulaAlarmes;
679     private javax.swing.JPanel jPanel1;
680     private javax.swing.JScrollPane jScrollPane1;
681     private javax.swing.JSeparator jSeparator1;
682     // End of variables declaration
683
684 }
685 }

```

### Codi D.17: (AppSeguiment) SplashScreen.java

```

1  package Vista;
2
3
4  import java.awt.Dimension;
5  import java.awt.Toolkit;
6  import javax.swing.ImageIcon;
7  import javax.swing.JLabel;
8  import javax.swing.JWindow;
9
10 /**
11  * Classe que defineix la finestra Splash d'inici de l'aplicació
12  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
13  * @version 1.0 (Juny de 2010)
14  */
15 public class SplashScreen extends JWindow implements Runnable {
16
17     private GUI V;
18
19     /**
20     * Crea una instància de SplashScreen
21     * @param V        Referència a la GUI principal

```



```

22     *
23     */
24     public SplashScreen(GUI V) {
25         this.V = V;
26     }
27
28     /**
29     * Codi principal del thread de SplashScreen
30     */
31     public void run() {
32         int width = 500;
33         int height = 250;
34         Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
35         int x = (screen.width-width)/2;
36         int y = (screen.height-height)/2;
37         setBounds(x,y,width,height);
38         add(new JLabel(new ImageIcon("res/splash.png")));
39         setVisible(true);
40         try { Thread.sleep(4000); } catch (Exception e) {}
41         setVisible(false);
42         V.setVisible(true);
43     }
44
45 }

```

#### Codi D.18: (AppSeguiment) Vista.java

```

1 package Vista;
2
3 import java.util.Observer;
4
5 /**
6  * Interfície d'una vista per a aplicar el patró MVC (Model-View-Controller)
7  * @author Oriol Ciurana (oriol.ciurana@entel.upc.edu)
8  * @version 1.0 (Juny de 2010)
9  */
10 public interface Vista extends Observer { }

```



## D.4 Guia de l'Usuari

# Guia d'Usuari

Versió 1.0



Juny de 2010

# Contingut

<b>1</b>	<b>Posada en funcionament</b>	<b>1</b>
1.1	Descàrrega de l'aplicació . . . . .	1
1.2	Instal·lació de l'aplicació . . . . .	1
1.3	Iniciar l'aplicació . . . . .	1
1.4	Zones de l'aplicació . . . . .	1
<b>2</b>	<b>Gestió de pacients</b>	<b>3</b>
2.1	Alta d'un pacient . . . . .	3
2.2	Modificar dades d'un pacient . . . . .	3
2.3	Eliminar un pacient . . . . .	4
<b>3</b>	<b>Obtenció de l'historial d'alarmes</b>	<b>5</b>
3.1	Procés d'obtenció . . . . .	5
3.2	Possibles errors . . . . .	5
<b>4</b>	<b>Zones de la pantalla principal</b>	<b>6</b>
4.1	Selector del pacient a analitzar . . . . .	6
4.2	Visualitzador de dades del pacient . . . . .	6
4.3	Taula d'alarmes . . . . .	6
4.4	Selector de tipus de gràfic . . . . .	7
4.5	Visualitzador de la gràfica estadística . . . . .	7
<b>5</b>	<b>Generació de l'informe PDF</b>	<b>8</b>
5.1	Procés de generació . . . . .	8
5.2	Estructura de l'informe . . . . .	8
<b>6</b>	<b>Notes sobre l'aplicació</b>	<b>9</b>
6.1	Requisits mínims . . . . .	9
6.2	Changelog . . . . .	9
6.3	Bugs i suggeriments . . . . .	9



## Posada en funcionament

### 1.1 Descàrrega de l'aplicació

Per a descarregar l'aplicació cal accedir a la pàgina web de PPSTOP <http://lewis.upc.es/ppstop> i anar a la secció **Aplicació de seguiment**. Allí trobareu l'enllaç de descàrrega de l'instal·lador multiplataforma.

### 1.2 Instal·lació de l'aplicació

Una vegada descarregada l'aplicació, la seva instal·lació es gestiona en tot moment a través d'un instal·lador d'aplicacions. En funció del vostre sistema operatiu, la instal·lació procedirà d'una o altra manera, deixant l'**aplicació de seguiment PPSTOP** a punt per a ser utilitzada.

### 1.3 Iniciar l'aplicació

Una vegada instal·lada l'aplicació, heu de procedir a iniciar l'aplicació fent clic a a icona de **PPStop seguiment** que us apareixerà al vostre menú de programes. Aquesta icona dóna accés a l'aplicació.



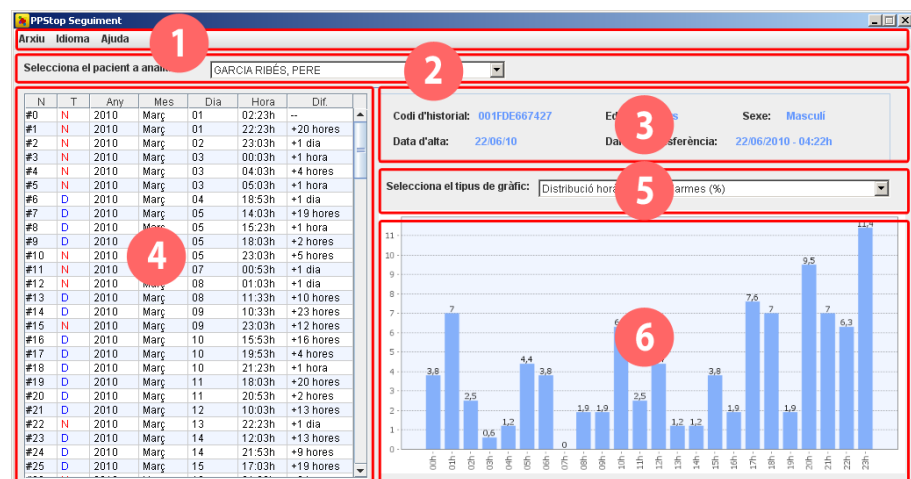
#### PPSTOP Seguiment

Tot seguit apareixerà la pantalla de benvinguda de l'aplicació, i posteriorment s'obrirà la pantalla principal de l'aplicació.



### 1.4 Zones de l'aplicació

La pantalla principal està dividida en diferents zones per a facilitar la tasca a l'usuari, cada una d'elles destinada a una finalitat diferent.











S'hi poden distingir les següents zones:

- 1 Menú principal d'accés a diferents opcions de l'aplicació.
- 2 Eina selectora de pacient a realitzar el seguiment.
- 3 Visualitzador de les dades del pacient seleccionat.
- 4 Taula amb l'historial d'alarmes del pacient seleccionat.
- 5 Eina selectora de tipus de gràfica a visualitzar.
- 6 Visualitzador de la gràfica estadística seleccionada.




El funcionament del programa és senzill i intuïtiu. A la zona central de la **pantalla principal** únicament hi ha selectors i visualitzadors de dades i gràfics, i aquesta característica no canvia en cap moment. La resta d'opcions del programa estan disponibles a través del **menú principal**.

Les opcions disponibles en el **menú principal** són les següents:

## MENÚ ARXIU:

-  Dóna d'alta un nou pacient a l'aplicació
-  Modifica les dades d'un pacient existent a l'aplicació
-  Elimina un pacient existent a l'aplicació
-  Obté l'historial del terminal PPStop del pacient
-  Genera un informe PDF amb l'historial i gràfiques estadístiques del pacient
-  Surt de l'aplicació

## MENÚ IDIOMA:

-  Canvia l'idioma de la interfície al català
-  Canvia l'idioma de la interfície al castellà
-  Canvia l'idioma de la interfície a l'anglès

## MENÚ AJUDA:

-  Mostra l'ajuda de l'aplicació



## Gestió de pacients

### 2.1 Alta d'un pacient

Per a donar d'alta un pacient a l'aplicació cal accedir a l'ítem **Nou pacient** del menú **Arxiu**. Aquest ítem obre una nova finestra d'introducció de les dades del nou pacient.

Els camps d'entrada sol·licitats, tots ells obligatoris, són els següents:

<b>Nom</b>	Nom del pacient
<b>Cognoms</b>	Cognoms del pacient
<b>Data de naixement</b>	Data de naixement del pacient
<b>Sexe</b>	Sexe del pacient
<b>Codi d'història</b>	Codi d'història del pacient

El paràmetre més important de tots és el **codi d'història**, que serà imprescindible per a que posteriorment l'aplicació pugui recuperar amb èxit l'història d'alarmes emmagatzemat en l'aplicació **PPSTOP** del terminal mòbil. Per tal d'esbrinar aquest paràmetre consulteu el manual de l'aplicació del terminal mòbil.

Una vegada completades totes les dades, cal pulsar el botó **Donar d'alta** i el nou pacient passarà a aparèixer al **selector de pacient** de la pantalla principal.

### 2.2 Modificar dades d'un pacient

Per a modificar les dades d'un pacient prèviament cal seleccionar-lo a través del **selector de pacient** de la pantalla principal. Una vegada seleccionat, ucal accedir a l'ítem **Modificar pacient** del menú **Arxiu**. Aquest ítem obre una nova finestra de modificació de les dades del pacient similar a la finestra de **Nou usuari** vist anteriorment.





Modificar pacient

Modifiqueu les dades del pacient:

Nom: Pere

Cognoms: Garcia Ribés

Data de naixement: 14/05/2005 (dd/mm/yyyy)

Sexe: Masculí

Codi d'historial: 001FDE667427

Modificar dades

És important veure que el camp **codi d'historial** no és modificable. Això és així perquè aquest camp us identifica de manera única al pacient i les seves alarmes de l'historial dins de l'aplicació.

Una vegada modificades les dades desitjades, cal pulsar el botó **Modificar dades**.

## 2.3 Eliminar un pacient

Per a eliminar un pacient prèviament cal seleccionar-lo a través del **selector de pacient** de la pantalla principal. Una vegada seleccionat, ucal accedir a l'ítem **Eliminar pacient** del menú **Arxiu**. Prèviament a l'eliminació del pacient, l'aplicació mostra un missatge de confirmació de l'acció que cal respondre afirmativament.

PPStop Seguiment

Segur que vols eliminar el pacient: GARCIA RIBÉS, PERE ?

Sí No

L'eliminació d'un pacient implica que l'aplicació també esborri el seu historial d'alarmes, de manera que és tracta d'una operació irreversible.



## Obtenció de l'historial d'alarmes

### 3.1 Procés d'obtenció

Per a obtenir l'historial d'alarmes d'un pacient prèviament cal seleccionar-lo a través del **selector de pacient** de la pantalla principal. Una vegada seleccionat, cal accedir a l'ítem **Obtenir historial** del menú **Arxiu**. Per a l'obtenció de l'historial és indispensable disposar de connectivitat **Bluetooth** a l'ordinador i tenir oberta l'aplicació **PPSTOP** del terminal mòbil, i aquest últim amb la connectivitat **Bluetooth** activada.



Abans de procedir a l'obtenció de l'historial, l'aplicació mostra una alerta informant que el procés pot trigar uns segons, i que cal mantenir-se a l'espera. Si l'operació s'ha realitzat amb èxit, l'aplicació mostra una nova alerta a tal efecte on s'indica el nombre d'alarmes que s'han obtingut, i aquestes automàticament són associades al pacient i incorporades en les dades que es mostren a la pantalla principal. En cas que l'operació no s'hagi realitzat amb èxit, l'aplicació mostra un missatge d'error indicant-ne el motiu.

És important aclarir que en obtenir l'historial, aquelles alarmes que ja han estat obtingudes en una transferència prèvia (repeticions) no es tornen a afegir i s'ignoren. Igualment, en cas que l'historial obtingut no incorpori alarmes obtingudes en una transferència prèvia, aquestes romanen en l'historial del pacient, és a dir, no s'eliminen.

### 3.2 Possibles errors

Els possibles errors que poden sorgir durant l'obtenció d'un l'historial i la seva manera de procedir són:

<b>Manca de connectivitat Bluetooth</b>	Cal revisar el Bluetooth estigui actiu
<b>No s'ha trobat el servei PPSTOP</b>	Cal revisar que el terminal PPSTOP tingui l'aplicació PPSTOP en execució, el Bluetooth actiu, i que els codis d'historial coincideixin.
<b>Error en la comunicació</b>	Cal reintentar l'operació una altra vegada



## Zones de la pantalla principal

### 4.1 Selector del pacient a analitzar

Aquest selector mostra tots els pacients registrats a l'aplicació i permet seleccionar-ne un, ja sigui per a fer-ne el seu seguiment a través de la pantalla principal, o bé per modificar les seves dades o eliminar-lo, o bé per a obtenir el seu historial d'alarmes i/o generar l'informe de seguiment en PDF.

### 4.2 Visualitzador de dades del pacient

Aquest visualitzador mostra les dades personals del pacient seleccionat i alguna dada extra d'interès.

<b>Codi d'historial</b>	Codi d'historial del pacient
<b>Edat</b>	Edat actual del pacient
<b>Sexe</b>	Sexe del pacient
<b>Data d'alta</b>	Data d'alta del pacient a l'aplicació
<b>Darrera transferència</b>	Data de la darrera obtenció d'historial del pacient

### 4.3 Taula d'alarmes

Aquesta taula mostra amb tot detall l'**historial d'alarmes** del pacient, des de la primera alarma continguda a l'historial fins a la última i totes elles ordenades cronològicament.

N	T	Any	Mes	Dia	Hora	Dif.
#1	N	2010	Març	01	02:23h	--
#2	N	2010	Març	01	22:23h	+20 hores
#3	N	2010	Març	02	23:03h	+1 dia
#4	N	2010	Març	03	00:03h	+1 hora
#5	N	2010	Març	03	04:03h	+4 hores
#6	N	2010	Març	03	05:03h	+1 hora
#7	D	2010	Març	04	18:53h	+1 dia
#8	D	2010	Març	05	14:03h	+19 hores
#9	D	2010	Març	05	15:23h	+1 hora
#10	D	2010	Març	05	18:03h	+2 hores
#11	N	2010	Març	05	23:03h	+5 hores
#12	N	2010	Març	07	00:53h	+1 dia
#13	N	2010	Març	08	01:03h	+1 dia
#14	D	2010	Març	08	11:33h	+10 hores
#15	D	2010	Març	09	10:33h	+23 hores
#16	N	2010	Març	09	23:03h	+12 hores
#17	D	2010	Març	10	15:53h	+16 hores
#18	D	2010	Març	10	19:53h	+4 hores
#19	D	2010	Març	10	21:23h	+1 hora
#20	D	2010	Març	11	18:03h	+20 hores
#21	D	2010	Març	11	20:53h	+2 hores
#22	D	2010	Març	12	10:03h	+13 hores
#23	N	2010	Març	13	22:23h	+1 dia
#24	D	2010	Març	14	12:03h	+13 hores
#25	D	2010	Març	14	21:53h	+9 hores
#26	D	2010	Març	15	17:03h	+19 hores

Els camps mostrats en aquesta taula són els següents:

<b>N (número)</b>	Número d'alarma
<b>T (ipus)</b>	Tipus d'alarma: Diürna(D)/Nocturna(N)
<b>Any</b>	Any de l'alarma
<b>Mes</b>	Mes de l'alarma
<b>Dia</b>	Dia de l'alarma
<b>Hora</b>	Hora de l'alarma (format 24h)
<b>Dif. (diferència)</b>	Diferència temporal amb l'alarma anterior

#### 4.4 Selector de tipus de gràfic

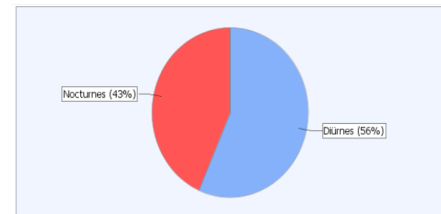
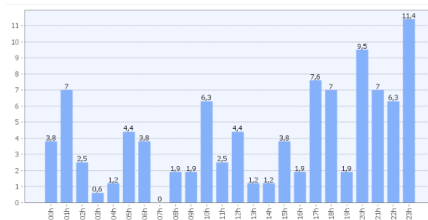
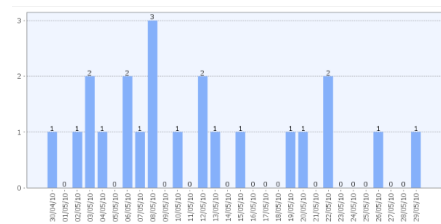
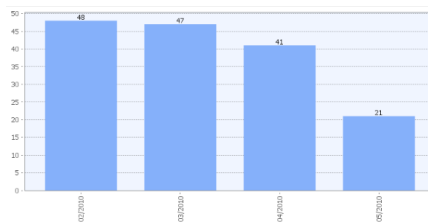
Aquest selector mostra tots els possibles **gràfics estadístics** que es poden visualitzar en el visualitzador de gràfics.

Hi ha 4 possibles gràfics disponibles, que són els següents:

- 1 Gràfica de barres de l'evolució mensual del nombre d'alarmes.
- 2 Gràfica de barres de l'evolució del nombre d'alarmes dels darrers 30 dies.
- 3 Gràfica de barres de distribució horària de les alarmes (%).
- 4 Gràfica circular de distribució (diürna/nocturna) de les alarmes.

#### 4.5 Visualitzador de la gràfica estadística

Aquest visualitzador mostra la gràfica seleccionada pel **selector de tipus de gràfic**. A continuació es mostren uns gràfics d'exemple corresponents, respectivament, als 4 tipus de gràfiques esmentades anteriorment.



Les gràfiques mostren una representació estadística de les alarmes presents fins al moment en l'**historial d'alarmes**.



## Generació de l'informe PDF

### 5.1 Procés de generació

Per a generar un informe portable en PDF d'un pacient prèviament cal seleccionar-lo a través del **selector de pacient** de la pantalla principal. Una vegada seleccionat, cal accedir a l'ítem **Generar informe PDF** del menú **Arxiu**.

Prèviament a la generació de l'informe, l'aplicació mostra una finestra on permet que l'usuari seleccioni el nom del fitxer i la destinació (path) on aquest serà generat. Només es poden generar informes d'aquells pacients que el seu historial tingui alarmes, altrament l'aplicació mostra un missatge d'error alertant de tal condició.

### 5.2 Estructura de l'informe

L'informe PDF generat mostra les mateixes dades i gràfics que els mostrats en la pantalla principal de l'aplicació, i en el mateix idioma que l'establert en l'aplicació. Aquesta informació s'organitza en seccions consecutives:

1

Capçalera de PPSTOP, on s'indica la data de generació de l'informe.

2

Dades personals del pacient i dades d'interès: data d'alta i data de la darrera transferència.

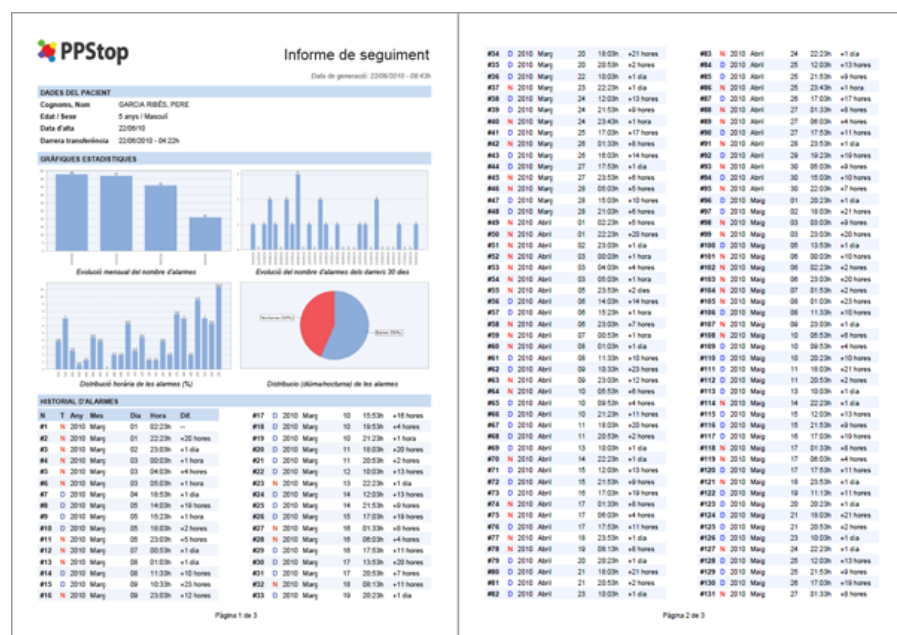
3

Gràfiques estadístiques.

4

Historial d'alarmes.

L'informe generat pot ser de més d'una pàgina en funció de la longitud de l'historial d'alarmes.





## Notes sobre l'aplicació

### 6.1 Requisits mínims

Els requisits mínims d'un ordinador de sobretaula per al correcte funcionament de l'aplicació **PPSTOP Seguiment** són els següents:

- Connectivitat Bluetooth
- JRE (Java Runtime Environment)
- 200KB d'espai lliure en memòria
- Resolució mínima de pantalla de 800px x 600px

### 6.2 Changelog

**Versió 1.0** (06/2010)

· Primera versió funcional

### 6.3 Bugs i suggeriments

Aquesta versió de l'aplicació està en fase de proves. Es pot reportar qualsevol errada i/o deficiència que s'hi trobi, i també fer suggeriments per a millorar-la enviant un correu a la següent adreça de correu electrònic:

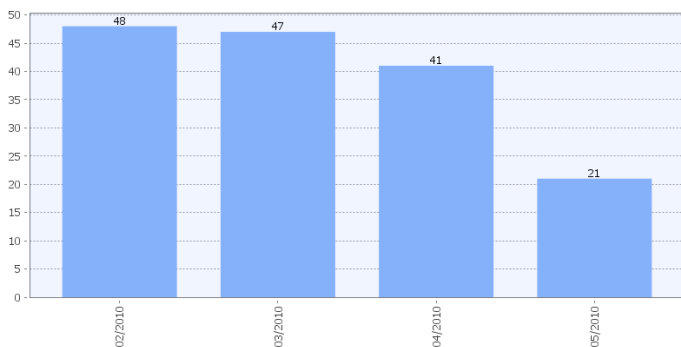
**Oriol Ciurana** [oriol.ciurana@entel.upc.edu](mailto:oriol.ciurana@entel.upc.edu)

## D.5 Exemple d'informe

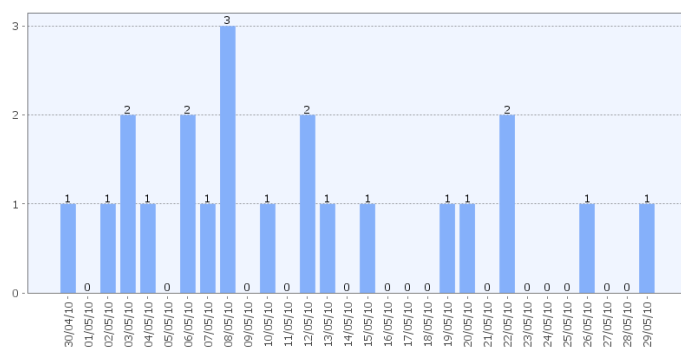
## DADES DEL PACIENT

**Cognoms, Nom** GARCIA RIBÉS, PERE  
**Edat / Sexe** 5 anys / Masculí  
**Data d'alta** 22/06/10  
**Darrera transferència** 22/06/2010 - 04:22h

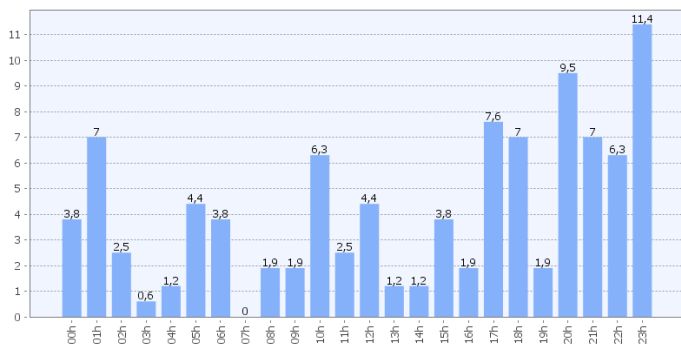
## GRÀFIQUES ESTADÍSTIQUES



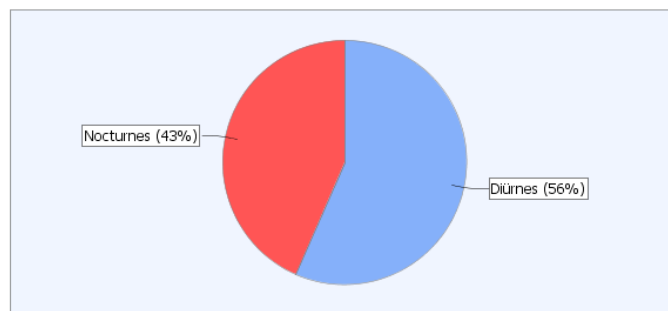
Evolució mensual del nombre d'alarmes



Evolució del nombre d'alarmes dels darrers 30 dies



Distribució horària de les alarmes (%)



Distribució (diürna/nocturna) de les alarmes

## HISTORIAL D'ALARMES

N	T	Any	Mes	Dia	Hora	Dif.
#1	N	2010	Març	01	02:23h	--
#2	N	2010	Març	01	22:23h	+20 hores
#3	N	2010	Març	02	23:03h	+1 dia
#4	N	2010	Març	03	00:03h	+1 hora
#5	N	2010	Març	03	04:03h	+4 hores
#6	N	2010	Març	03	05:03h	+1 hora
#7	D	2010	Març	04	18:53h	+1 dia
#8	D	2010	Març	05	14:03h	+19 hores
#9	D	2010	Març	05	15:23h	+1 hora
#10	D	2010	Març	05	18:03h	+2 hores
#11	N	2010	Març	05	23:03h	+5 hores
#12	N	2010	Març	07	00:53h	+1 dia
#13	N	2010	Març	08	01:03h	+1 dia
#14	D	2010	Març	08	11:33h	+10 hores
#15	D	2010	Març	09	10:33h	+23 hores
#16	N	2010	Març	09	23:03h	+12 hores

#17	D	2010	Març	10	15:53h	+16 hores
#18	D	2010	Març	10	19:53h	+4 hores
#19	D	2010	Març	10	21:23h	+1 hora
#20	D	2010	Març	11	18:03h	+20 hores
#21	D	2010	Març	11	20:53h	+2 hores
#22	D	2010	Març	12	10:03h	+13 hores
#23	N	2010	Març	13	22:23h	+1 dia
#24	D	2010	Març	14	12:03h	+13 hores
#25	D	2010	Març	14	21:53h	+9 hores
#26	D	2010	Març	15	17:03h	+19 hores
#27	N	2010	Març	16	01:33h	+8 hores
#28	N	2010	Març	16	06:03h	+4 hores
#29	D	2010	Març	16	17:53h	+11 hores
#30	D	2010	Març	17	13:53h	+20 hores
#31	D	2010	Març	17	20:53h	+7 hores
#32	N	2010	Març	18	08:13h	+11 hores
#33	D	2010	Març	19	20:23h	+1 dia



#34	D	2010	Març	20	18:03h	+21 hores
#35	D	2010	Març	20	20:53h	+2 hores
#36	D	2010	Març	22	10:03h	+1 dia
#37	N	2010	Març	23	22:23h	+1 dia
#38	D	2010	Març	24	12:03h	+13 hores
#39	D	2010	Març	24	21:53h	+9 hores
#40	N	2010	Març	24	23:43h	+1 hora
#41	D	2010	Març	25	17:03h	+17 hores
#42	N	2010	Març	26	01:33h	+8 hores
#43	D	2010	Març	26	16:03h	+14 hores
#44	D	2010	Març	27	17:53h	+1 dia
#45	N	2010	Març	27	23:53h	+6 hores
#46	N	2010	Març	28	05:03h	+5 hores
#47	D	2010	Març	28	15:03h	+10 hores
#48	D	2010	Març	28	21:03h	+6 hores
#49	N	2010	Abril	01	02:23h	+5 hores
#50	N	2010	Abril	01	22:23h	+20 hores
#51	N	2010	Abril	02	23:03h	+1 dia
#52	N	2010	Abril	03	00:03h	+1 hora
#53	N	2010	Abril	03	04:03h	+4 hores
#54	N	2010	Abril	03	05:03h	+1 hora
#55	N	2010	Abril	05	23:53h	+2 dies
#56	D	2010	Abril	06	14:03h	+14 hores
#57	D	2010	Abril	06	15:23h	+1 hora
#58	N	2010	Abril	06	23:03h	+7 hores
#59	N	2010	Abril	07	00:53h	+1 hora
#60	N	2010	Abril	08	01:03h	+1 dia
#61	D	2010	Abril	08	11:33h	+10 hores
#62	D	2010	Abril	09	10:33h	+23 hores
#63	N	2010	Abril	09	23:03h	+12 hores
#64	N	2010	Abril	10	05:53h	+6 hores
#65	D	2010	Abril	10	09:53h	+4 hores
#66	D	2010	Abril	10	21:23h	+11 hores
#67	D	2010	Abril	11	18:03h	+20 hores
#68	D	2010	Abril	11	20:53h	+2 hores
#69	D	2010	Abril	13	10:03h	+1 dia
#70	N	2010	Abril	14	22:23h	+1 dia
#71	D	2010	Abril	15	12:03h	+13 hores
#72	D	2010	Abril	15	21:53h	+9 hores
#73	D	2010	Abril	16	17:03h	+19 hores
#74	N	2010	Abril	17	01:33h	+8 hores
#75	N	2010	Abril	17	06:03h	+4 hores
#76	D	2010	Abril	17	17:53h	+11 hores
#77	N	2010	Abril	18	23:53h	+1 dia
#78	N	2010	Abril	19	08:13h	+8 hores
#79	D	2010	Abril	20	20:23h	+1 dia
#80	D	2010	Abril	21	18:03h	+21 hores
#81	D	2010	Abril	21	20:53h	+2 hores
#82	D	2010	Abril	23	10:03h	+1 dia

#83	N	2010	Abril	24	22:23h	+1 dia
#84	D	2010	Abril	25	12:03h	+13 hores
#85	D	2010	Abril	25	21:53h	+9 hores
#86	N	2010	Abril	25	23:43h	+1 hora
#87	D	2010	Abril	26	17:03h	+17 hores
#88	N	2010	Abril	27	01:33h	+8 hores
#89	N	2010	Abril	27	06:03h	+4 hores
#90	D	2010	Abril	27	17:53h	+11 hores
#91	N	2010	Abril	28	23:53h	+1 dia
#92	D	2010	Abril	29	19:23h	+19 hores
#93	N	2010	Abril	30	05:03h	+9 hores
#94	D	2010	Abril	30	15:03h	+10 hores
#95	N	2010	Abril	30	22:03h	+7 hores
#96	D	2010	Maig	01	20:23h	+1 dia
#97	D	2010	Maig	02	18:03h	+21 hores
#98	N	2010	Maig	03	03:03h	+9 hores
#99	N	2010	Maig	03	23:03h	+20 hores
#100	D	2010	Maig	05	13:53h	+1 dia
#101	N	2010	Maig	06	00:03h	+10 hores
#102	N	2010	Maig	06	02:23h	+2 hores
#103	N	2010	Maig	06	23:03h	+20 hores
#104	N	2010	Maig	07	01:53h	+2 hores
#105	N	2010	Maig	08	01:03h	+23 hores
#106	D	2010	Maig	08	11:33h	+10 hores
#107	N	2010	Maig	09	23:03h	+1 dia
#108	N	2010	Maig	10	05:53h	+6 hores
#109	D	2010	Maig	10	09:53h	+4 hores
#110	D	2010	Maig	10	20:23h	+10 hores
#111	D	2010	Maig	11	18:03h	+21 hores
#112	D	2010	Maig	11	20:53h	+2 hores
#113	D	2010	Maig	13	10:03h	+1 dia
#114	N	2010	Maig	14	22:23h	+1 dia
#115	D	2010	Maig	15	12:03h	+13 hores
#116	D	2010	Maig	15	21:53h	+9 hores
#117	D	2010	Maig	16	17:03h	+19 hores
#118	N	2010	Maig	17	01:33h	+8 hores
#119	N	2010	Maig	17	06:03h	+4 hores
#120	D	2010	Maig	17	17:53h	+11 hores
#121	N	2010	Maig	18	23:53h	+1 dia
#122	D	2010	Maig	19	11:13h	+11 hores
#123	D	2010	Maig	20	20:23h	+1 dia
#124	D	2010	Maig	21	18:03h	+21 hores
#125	D	2010	Maig	21	20:53h	+2 hores
#126	D	2010	Maig	23	10:03h	+1 dia
#127	N	2010	Maig	24	22:23h	+1 dia
#128	D	2010	Maig	25	12:03h	+13 hores
#129	D	2010	Maig	25	21:53h	+9 hores
#130	D	2010	Maig	26	17:03h	+19 hores
#131	N	2010	Maig	27	01:33h	+8 hores

#132	D	2010	Maig	27	09:03h	+7 hores
#133	D	2010	Maig	27	17:53h	+8 hores
#134	D	2010	Maig	28	21:53h	+1 dia
#135	D	2010	Maig	29	19:23h	+21 hores
#136	N	2010	Maig	30	05:03h	+9 hores
#137	D	2010	Juny	02	20:23h	+2 dies
#138	D	2010	Juny	03	18:03h	+21 hores
#139	N	2010	Juny	03	23:03h	+5 hores
#140	D	2010	Juny	04	20:53h	+21 hores
#141	D	2010	Juny	06	10:03h	+1 dia
#142	N	2010	Juny	06	22:23h	+12 hores
#143	D	2010	Juny	07	12:03h	+13 hores
#144	N	2010	Juny	08	01:53h	+13 hores
#145	N	2010	Juny	08	06:03h	+4 hores
#146	D	2010	Juny	08	21:33h	+15 hores
#147	N	2010	Juny	10	06:03h	+1 dia
#148	D	2010	Juny	12	15:53h	+2 dies
#149	N	2010	Juny	12	23:53h	+8 hores
#150	N	2010	Juny	13	08:13h	+8 hores
#151	D	2010	Juny	15	16:23h	+2 dies
#152	D	2010	Juny	19	18:03h	+4 dies
#153	D	2010	Juny	20	20:53h	+1 dia
#154	D	2010	Juny	22	10:03h	+1 dia
#155	D	2010	Juny	22	16:23h	+6 hores
#156	N	2010	Juny	26	02:23h	+3 dies
#157	N	2010	Juny	29	00:53h	+2 dies

# Acrònims

<b>ACL</b> Asynchronous Connection Less	<b>GPRS</b> General Packet Radio Service
<b>AMR</b> Adaptative Multi Rate	<b>GSM</b> Global System for Mobile Communications
<b>AMS</b> Application Management Software	<b>GUI</b> Graphical User Interface
<b>API</b> Application Programming Interface	<b>HTTP</b> Hypertext Transfer Protocol
<b>ARQ</b> Automatic Retransmission Query	<b>IEEE</b> Institution of Electrical Engineers
<b>AWT</b> Abstract Window Toolkit	<b>JABWT</b> Java API for Bluetooth Wireless Technology
<b>CDC</b> Connected Device Configuration	<b>JCP</b> Java Community Process
<b>CLDC</b> Connected Limited Device Configuration	<b>JDBC</b> Java DataBase Connectivity
<b>COMMAPI</b> Java Communications API	<b>JDK</b> Java Development Kit
<b>CRC</b> Cyclic Redundancy Check	<b>JSR</b> Java Specification Requests
<b>CSVD</b> Continously Variable Slope Delta	<b>JVM</b> Java Virtual Machine
<b>CVM</b> Compart Virtual Machine	<b>J2EE</b> Java 2 Enterprise Edition
<b>DAC</b> Device Access Code	<b>J2SE</b> Java 2 Standard Edition
<b>DDP</b> Device Discovery Protocol	<b>J2ME</b> Java 2 Micro Edition
<b>DIAC</b> Dedicated Inquiry Access Code	<b>HCI</b> Host Controller Interface
<b>EDR</b> Enhanced Data Rate	<b>HTTP</b> Hypertext Transfer Protocol
<b>EDT</b> Event Dispatcher Thread	<b>IDE</b> Integrated Development Environment
<b>ENMP</b> Enuresi Nocturna Monosintomàtica Primària	<b>IEC</b> International Electrotechnical Commission
<b>FEC</b> Forward Error Correction	<b>IEEE</b> Institute of Electrical and Electronics Engineers
<b>FH</b> Frequency Hop	<b>IrDA</b> Official Infrared Data Association
<b>FHS</b> Frequency Hop Synchronization	<b>ISO</b> International Organization for Standardization
<b>FHSS</b> Frequency Hoping Spread Spectrum	<b>ISM</b> Industrial Scientific and Medical
<b>FDD</b> Frequency Division Duplex	<b>JAR</b> Java ARchive
<b>GAP</b> Generic Access Profile	<b>JAD</b> Java Application Descriptor
<b>GCF</b> Generic Connection Framework	
<b>GFSK</b> Gaussian Frequency Shift Keying	
<b>GIAC</b> General Inquiry Access Code	

<b>JCP</b> Java Community Process	<b>SCO</b> Synchronous Connection Oriented
<b>JDK</b> Java Development Kit	<b>SDDB</b> Service Discovery Database
<b>J2EE</b> Java 2 Enterprise Edition	<b>SDK</b> Software Development Kit
<b>J2ME</b> Java 2 Micro Edition	<b>SDP</b> Service Discovery Protocol
<b>J2SDK</b> Java 2 Standard Development Kit	<b>SIG</b> Special Interest Group
<b>J2SE</b> Java 2 Standard Edicion	<b>SIM</b> Subscriber Identity Module
<b>JMF</b> Java Media Framework	<b>SMS</b> Short Message Service
<b>JVM</b> Java Virtual Machine	<b>SPP</b> Serial Port Profile
<b>KVM</b> Kilobyte Virtual Machine	<b>SSL</b> Secure Socket Layer
<b>LAN</b> Local Area Network	<b>TCP</b> Transmission Control Protocol
<b>LMP</b> Link Manager Protocol	<b>TCS</b> Telephony Control Protocol
<b>LWUIT</b> Lightweight User Interface Toolkit	<b>TIC</b> Tecnologies de la Informació i la Comunicació
<b>L2CAP</b> Logical Link Control and Adaption Protocol	<b>TDD</b> Time Division Duplex
<b>MAC</b> Media Access Control	<b>UC</b> Unitat Control
<b>MIDP</b> Mobile Information Device Profile	<b>UART</b> Universal Asynchronous Receiver/-Transmitter
<b>MMAPI</b> Mobile Media API	<b>UMTS</b> Universal Mobile Telecommunications System
<b>MVC</b> Model View Controller	<b>URI</b> Uniform Resource Identifier
<b>OBEX</b> OBject EXchange	<b>URL</b> Uniform Resource Locator
<b>OEM</b> Original Equipment Manufacturer	<b>USB</b> Universal Serial Bus
<b>OTA</b> Over-the-Air	<b>UUID</b> Universal Unique IDentifier
<b>PAN</b> Personal Area Network	<b>UWB</b> Ultra Wide Band
<b>PC</b> Personal Computer	<b>VFIR</b> Very Fast Infrared
<b>PDA</b> Personal Digital Assistant	<b>WAP</b> Wireless Application Protocol
<b>PDF</b> Portable Document Format	<b>WAV</b> Waveform Audio Format
<b>PIN</b> Personal Identification Number	<b>WDP</b> WAP Datagram Protocol
<b>PKI</b> Public Key Infrastructure	<b>Wi-Fi</b> Wireless Fidelity
<b>QoS</b> Quality of Service	<b>WLAN</b> Wireless Local Area Network
<b>RFCOMM</b> Radio Frequency Communication	<b>WPAN</b> Wireless Personal Area Network
<b>RAM</b> Random Access Memory	<b>WWW</b> World Wide Web
<b>RMS</b> Record Management System	<b>W-USB</b> Certified Wireless Universal Serial Bus
<b>ROM</b> Read-Only Memory	
<b>RSA</b> Rivest Shamir Adleman	

# Bibliografia

- [1] Wikipedia. Enuresi. <http://ca.wikipedia.org/wiki/Enuresi>.
- [2] Fundació Sant Joan de Déu. Dr Santiago García-Tornel. L'enuresi i el seu tractament. [http://www.fsjd.org/cat/ic2\\_noticies.php?art\\_id=26&idioma=1](http://www.fsjd.org/cat/ic2_noticies.php?art_id=26&idioma=1).
- [3] Dr Santiago García-Tornel. Utilidad de las alarmas nocturnas en la enuresis. [http://www.sepeap.org/imagenes/secciones/Image/\\_USER\\_/TS\\_Utilidad\\_alarmas\\_nocturnas\\_enuresis.pdf](http://www.sepeap.org/imagenes/secciones/Image/_USER_/TS_Utilidad_alarmas_nocturnas_enuresis.pdf).
- [4] Bluetooth SIG. Bluetooth SIG Membership Website. <http://www.bluetooth.com>.
- [5] Bluetooth SIG. Specification of the Bluetooth System. Volume 0: Core Specification v2.1 + EDR. [http://www.bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA-E5535258B056/6545/Core\\_V21\\_\\_EDR.zip](http://www.bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA-E5535258B056/6545/Core_V21__EDR.zip), 2007.
- [6] Sun Microsystems. The Java 2 Platform Micro Edition overview (J2ME). <http://java.sun.com/j2me>.
- [7] Sun Microsystems. Using the Java APIs for Bluetooth Wireless Technology. Part 1 - API Overview. <http://developers.sun.com/mobility/apis/articles/bluetoothintro/>, 2004.
- [8] Sun Microsystems. Using the Java APIs for Bluetooth Wireless Technology. Part 2 - Putting the Core APIs to Work. <http://developers.sun.com/mobility/apis/articles/bluetoothcore/>, 2004.
- [9] Sun Microsystems. The J2ME Mobile Media API. <http://developers.sun.com/mobility/midp/articles/mmapioverview/>, 2003.
- [10] Sun Microsystems. LWUIT Project Home. <https://lwuit.dev.java.net/>.
- [11] Sun Microsystems. The Swing Tutorial: Creating a GUI With JFC/Swing. <http://java.sun.com/docs/books/tutorial/uiswing/>, 2009.
- [12] Wikipedia. Adaptive Multi-Rate Audio Codec. [http://en.wikipedia.org/wiki/Adaptive\\_Multi-Rate\\_audio\\_codec](http://en.wikipedia.org/wiki/Adaptive_Multi-Rate_audio_codec).
- [13] Nokia. Nokia Multimedia Converter 2.0. [http://www.forum.nokia.com/info/sw.nokia.com/id/d1c17a7f-1231-4385-8c17-04f28f4f2d8e/Nokia\\_Multimedia\\_Converter\\_2.0.html](http://www.forum.nokia.com/info/sw.nokia.com/id/d1c17a7f-1231-4385-8c17-04f28f4f2d8e/Nokia_Multimedia_Converter_2.0.html), 2003.

- [14] Wikipedia. WAP and WAP-Push. [http://en.wikipedia.org/wiki/Wireless\\_Application\\_Protocol](http://en.wikipedia.org/wiki/Wireless_Application_Protocol).
- [15] Keijo Kasvi. The SMS Server Tools Project Homepage. <http://smstools3.kekekasvi.com/>.
- [16] Sun Microsystems. The Java 2 Platform, Standard Edition overview (J2SE). <http://java.sun.com/j2se>.
- [17] Bluecove. BlueCove Java library for Bluetooth (JSR-82 implementation). <http://www.bluecove.org/>.
- [18] Avetana. Avetana Bluetooth JSR-82 implementation. <http://www.avetana-gmbh.de>.
- [19] Sun Microsystems. JavaDB/Derby Technology Database. <http://developers.sun.com/javadb/>.
- [20] JFree free software for Java. JFreeChart Java Chart Library. <http://www.jfree.org/jfreechart/>.
- [21] iText. iText PDF, a F/OSS Java-PDF Library. <http://www.itextpdf.com>.
- [22] BitRock. BitRock InstallBuilder: Easy to Use Multiplatform Installers and Customized Open Source Stacks. <http://www.bitrock.com/>.
- [23] Nathan J Muller. *Tecnología Bluetooth*. Madrid, McGraw-Hill, 2002.
- [24] IEEE 802 LAN/MAN Standards Committee. IEEE 802.15 WPAN Task Group 1 (TG1). <http://www.ieee802.org/15/pub/TG1.html>.
- [25] Bluetooth SIG. Specification of the Bluetooth System. Volume 4: HCI Transports. [http://bluetooth.com/NR/rdonlyres/24C24181-5B5D-4566-BB4F-D4C302910B9D/1766/Volume4\\_SPEC.pdf](http://bluetooth.com/NR/rdonlyres/24C24181-5B5D-4566-BB4F-D4C302910B9D/1766/Volume4_SPEC.pdf), 2006.
- [26] Wikipedia. WLAN - Wireless LAN. <http://en.wikipedia.org/wiki/WLAN>.
- [27] Wikipedia. IrDA - Infrared Data Association. [http://en.wikipedia.org/wiki/Infrared\\_Data\\_Association](http://en.wikipedia.org/wiki/Infrared_Data_Association).
- [28] Wikipedia. W-USB - Wireless USB. [http://en.wikipedia.org/wiki/Wireless\\_USB](http://en.wikipedia.org/wiki/Wireless_USB).
- [29] Wikipedia. ZigBee. <http://en.wikipedia.org/wiki/ZigBee>.
- [30] Sun Microsystems. The Connected Device Configuration (CDC); JSR 36, JSR 218. <http://java.sun.com/products/cdc/>.
- [31] Sun Microsystems. The Connected Limited Device Configuration (CLDC); JSR 139. <http://java.sun.com/products/cldc/>.

- [32] Sun Microsystems. The Mobile Information Device Profile (MIDP); JSR 118. <http://java.sun.com/products/midp/>.
- [33] John W Muchow. *Core J2ME technology and MIDP*. Upper Saddle River (N.J.) Prentice Hall PTR, 2002.
- [34] Lucas Galvez, Sergio; Ortega. *Java a tope: J2ME*. Universidad de Málaga, ETSII, 2005.
- [35] Sun Microsystems. Databases and MIDP: Understanding the Record Management System. <http://developers.sun.com/mobility/midp/articles/databasesrms/>, 2004.
- [36] IBM. J2ME record management store: Add data storage capacities to your MIDlet apps. <http://www.ibm.com/developerworks/library/wi-rms/>, 2002.
- [37] Sun Microsystems. Package javax.microedition.rms. <http://java.sun.com/javame/reference/apis/jsr118/javax/microedition/rms/package-summary.html>.
- [38] J2ME Polish. Serialization: How to make you classes serializable. <http://www.j2mepolish.org/cms/leftsection/documentation/programming/serialization.html>.
- [39] Antonio J. (ESI University of Sevilla) Sierra. Recursive and non-recursive method to serialize object at J2ME. <http://www.w3.org/2006/02/Sierra10022006.pdf>, 2006.
- [40] Ranjith Hopkins, Bruce; Antony. *Bluetooth for Java*. Berkeley, CA : APress, 2003.
- [41] Pedro Daniel Campo, Celeste; Borches. *Java 2 Micro Edition: Soporte Bluetooth*. Universidad Carlos III de Madrid, 2004.
- [42] Kirk Virkus, Robert; Bateman. *Pro J2ME Polish : open source wireless Java tools suite*. Berkeley, CA Apress, 2005.
- [43] Java Community Process. Java Specification Requests for JSR-82. <http://jcp.org/en/jsr/detail?id=82>.
- [44] Sun Microsystems. LWUIT Java API Documentation (JavaDoc). <https://lwuit.dev.java.net/nonav/javadocs/index.html>, 2008.
- [45] Larry Huang, Albert S; Rudolph. *Bluetooth essentials for programmers*. New York, Cambridge University, 2007.
- [46] Jonathan Li, Sing; Knudsen. *Beginning J2ME : from novice to professional*. [S.l.] Apress, 3rd edition, 2005.
- [47] Jim Yuan, Michael Juntao; Colson. *Enterprise J2ME : developing mobile Java applications*. Upper Saddle River, NJ Prentice Hall, 2004.

- [48] Paul J; Kumar C. Bala Thompson, Timothy J; Kline. *Bluetooth application programming with the Java APIs: essentials edition*. Burlington, MA Morgan Kaufmann pub, 2008.
- [49] JSR-82. Articles, Tutorials and Discussion Forums about Java Bluetooth Programming and JSR-82 Bluetooth API. <http://www.jsr82.com/>.
- [50] Nokia. MIDP: Bluetooth API Developer's Guide v2.0. [http://sw.nokia.com/id/125b7ff5-f2dd-4441-8cfe-59e23c006373/MIDP\\_Bluetooth\\_API\\_Developers\\_Guide\\_v2\\_0\\_en.pdf](http://sw.nokia.com/id/125b7ff5-f2dd-4441-8cfe-59e23c006373/MIDP_Bluetooth_API_Developers_Guide_v2_0_en.pdf), 2006.
- [51] Sun Microsystems. The javax.bluetooth package. JavaDoc. <http://java.sun.com/javame/reference/apis/jsr082/javax/bluetooth/package-summary.html>, 2006.
- [52] Alberto Gimeno. *JSR-82. Bluetooth desde Java*. Creative Commons, 2004.
- [53] Sun Microsystems. LWUIT Tutorial. <https://lwuit.dev.java.net/nonav/tutorial/index.html>, 2008.
- [54] Sun Microsystems. Lightweight UI Toolkit Developers Guide. [https://lwuit.dev.java.net/files/documents/8797/95067/file\\_95067.dat/LWUIT%20Developer\\_Guide.pdf](https://lwuit.dev.java.net/files/documents/8797/95067/file_95067.dat/LWUIT%20Developer_Guide.pdf), 2009.
- [55] StackOverflow. Playing Audio with J2ME. <http://stackoverflow.com/questions/473809/playing-audio-with-j2me>, 2009.
- [56] Nokia. Nokia Community for mobile application developers. <http://www.forum.nokia.com/>.
- [57] Sony Ericsson. Sony Ericsson Community for mobile application developers. <http://developer.sonyericsson.com>.
- [58] Netbeans. The Netbeans IDE Official Website. <http://www.netbeans.org/>.
- [59] Wikipedia. Model View Controller, an architectural pattern in computer software development. <http://en.wikipedia.org/wiki/Model-view-controller>.
- [60] IBM. The MVC design pattern in MIDP development. <http://www.ibm.com/developerworks/wireless/library/wi-arch6/>, 2004.
- [61] Motorola. Introduction of MVC structure in Java ME clients. [http://developer.motorola.com/docstools/articles/MVC\\_Structure\\_20060301.pdf](http://developer.motorola.com/docstools/articles/MVC_Structure_20060301.pdf), 2006.
- [62] Sun Microsystems. Java(tm) Communications API HomePage. <http://java.sun.com/products/javacomm/>, 2009.
- [63] Infineon Technologies. eBMU PBA31308/2 v1.01 Hardware Description (Revision 1.0). [http://www.infineon.com/dgdl/eUniStone\\_V11+01\\_UM\\_HD\\_Rev1+0.pdf?folderId=db3a3043136c9a8b01136ccf27ec0004&fileId=db3a30431ed1d7b2011f18daee9e1290](http://www.infineon.com/dgdl/eUniStone_V11+01_UM_HD_Rev1+0.pdf?folderId=db3a3043136c9a8b01136ccf27ec0004&fileId=db3a30431ed1d7b2011f18daee9e1290), 2009.



- [64] Infineon Technologies. eUniStone (BT 2.0 + EDR) PBA31308/2 Product Brief. <http://www.infineon.com/cms/en/product/channel.html?channel=db3a30431ed1d7b2011f038697c5736c>, 2008.
- [65] Infineon Technologies. eBMU PBA31308/2 SPP-AT Application Software Description (Revision 2.0). [http://www.infineon.com/dgdl/eBMU\\_SPP\\_AT\\_V11+01\\_V1+1\\_UM\\_SD\\_Rev2+0.pdf?folderId=db3a3043136c9a8b01136ccf27ec0004&fileId=db3a30431ed1d7b2011f18dc76861295](http://www.infineon.com/dgdl/eBMU_SPP_AT_V11+01_V1+1_UM_SD_Rev2+0.pdf?folderId=db3a3043136c9a8b01136ccf27ec0004&fileId=db3a30431ed1d7b2011f18dc76861295), 2008.
- [66] Infineon Technologies. XC800 (8-bit flash microcontrollers) Family Overview. [http://www.infineon.com/dgdl/XC800\\_family\\_overview.pdf?folderId=db3a304412b407950112b40c497b0af6&fileId=db3a304319c6f18c0119f20b125b4f41](http://www.infineon.com/dgdl/XC800_family_overview.pdf?folderId=db3a304412b407950112b40c497b0af6&fileId=db3a304319c6f18c0119f20b125b4f41), 2008.
- [67] Infineon Technologies. XC864 4KB Flash 8-bit Microcontroller Series Product Brief. [http://www.silica.com/fileadmin/02\\_Products/05\\_Product-News/06\\_MCU-MPU/IFX\\_XC864-pb.pdf](http://www.silica.com/fileadmin/02_Products/05_Product-News/06_MCU-MPU/IFX_XC864-pb.pdf), 2008.
- [68] Infineon Technologies. XC864 8-Bit Single Chip Controller v1.1 Data Sheet. [http://www.infineon.com/dgdl/XC864\\_ds\\_V1.1.pdf?folderId=db3a304412b407950112b40c497b0af6&fileId=db3a30431b3e89eb011b5ac7914822ac](http://www.infineon.com/dgdl/XC864_ds_V1.1.pdf?folderId=db3a304412b407950112b40c497b0af6&fileId=db3a30431b3e89eb011b5ac7914822ac), 2009.
- [69] Infineon Technologies. Easy Kit XC864 Development Tools. <http://www.infineon.com/cms/en/product/channel.html?channel=db3a30431b3e89eb011b5acd0d5522b2>, 2009.
- [70] ARM. KEIL uVision IDE overview. <http://www.keil.com/uvision/>.
- [71] Infineon Technologies. DAVe for the Infineon TriCore microcontroller Family. <http://www.infineon.com/cms/en/product/channel.html?channel=ff80808112ab681d0112ab6b5b2807eb>.
- [72] Flachmann und Heggelbacher. Docklight Homepage. <http://www.fuh-edv.de/docklight/>.
- [73] Infineon Technologies. XC86x EK Getting Started. Examples for XC864. <http://www.evalkits.com/files/XC86x%20EK%20Getting%20Started.pdf>, 2008.
- [74] KEIL. Infineon XC864-1FRI References for KEIL. <http://www.keil.com/dd/chip/4621.htm>, 2009.
- [75] EEMB Corporation. CR1/3N Lithium Manganese Dioxide Battery Brief Datasheet. <http://eemb.com/pdf/Li-MnO2/CR1-3N.pdf>, 2009.
- [76] Nokia. Nokia 6124 Classic. <http://www.nokia.es/productos/moviles/6124>.
- [77] Nokia. Nokia E50. <http://www.nokia.es/soporte/producto/e50>.
- [78] Apache Software Foundation. Apache Derby Technology Database. <http://db.apache.org/derby/>.

- [79] Wikipedia. JDBC - Java Database Connectivity. [http://en.wikipedia.org/wiki/Java\\_Database\\_Connectivity](http://en.wikipedia.org/wiki/Java_Database_Connectivity).
- [80] JFree free software for Java. JFreeChart (version 1.0.13) API Documentation. <http://www.jfree.org/jfreechart/api/javadoc/index.html>.
- [81] David Gilbert. JFreeChart Class Library (version 1.0.12) Developer Guide. <http://www.4shared.com/document/0p7zttTZ/jfreechart-1012-US.htm>, 2009.
- [82] JFree free software for Java. JFree JCommon. <http://www.jfree.org/jcommon/>.
- [83] Python. Python Programming Language. <http://www.python.org/>.
- [84] Wikipedia. Adobe Flash. [http://en.wikipedia.org/wiki/Adobe\\_Flash](http://en.wikipedia.org/wiki/Adobe_Flash).
- [85] Wikipedia. ActionScript. <http://es.wikipedia.org/wiki/ActionScript>.
- [86] Massachusetts Institute of Technology. MIDlet Signing and restricted APIs. <http://channel.media.mit.edu/ict4d/content/midlet-signing-and-restricted-apis>, 2008.
- [87] Sun Microsystems. JABWT for CLDC Users Guide v2.5.2. Chapter 6: Security and MIDlet Signing. <http://java.sun.com/products/sjwtoolkit/wtk2.5.2/docs/UserGuide-html/security.html>, 2007.
- [88] Javia. MIDlet Signing. <http://blog.javia.org/midlet-signing/>, 2007.
- [89] Wikipedia. Public Key Cryptography. [http://en.wikipedia.org/wiki/Public-key\\_cryptography](http://en.wikipedia.org/wiki/Public-key_cryptography).
- [90] Wikipedia. Public Key Certificate. [http://en.wikipedia.org/wiki/Public\\_key\\_certificate](http://en.wikipedia.org/wiki/Public_key_certificate).
- [91] Wikipedia. Digital Signature. [http://en.wikipedia.org/wiki/Digital\\_signature](http://en.wikipedia.org/wiki/Digital_signature).
- [92] JavaBlog. How MIDlet Signing is Killing J2ME. <http://javablog.co.uk/2007/08/09/how-midlet-signing-is-killing-j2me/>, 2007.